IMPROVING FREIGHT CONSOLIDATION NETWORKS USING IP-BASED LOCAL SEARCH

A Thesis
Presented to
The Academic Faculty

by

Kathleen A. Lindsey

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
December 2012

IMPROVING FREIGHT CONSOLIDATION NETWORKS USING IP-BASED LOCAL SEARCH

Approved by:

Professor Alan Erera, Advisor
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Martin Savelsbergh, Advisor
School of Mathematical and Physical Sciences
The University of Newcastle

Professor Chelsea C. White III
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Ellis Johnson
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Atalay Atasu
College of Management
Georgia Institute of Technology

Date Approved: 15 August 2012

In loving memory of my grandfathers,

Dr. Warren Fulton Abercrombie and The Reverend William David Leech.

ACKNOWLEDGEMENTS

There are many people that have been a part of making this dissertation possible. First, I would like to thank my advisors, Alan Erera and Martin Savelsbergh. I am eternally grateful for their patience and dedication as I matured as a researcher and learned what it really meant to conduct research at a level worthy of a Ph.D. dissertation. Without their guidance, motivation, and mentorship, I would not be the person that I am today. I would also like to thank Ellis Johnson, Chip White, and Atalay Atasu for their willingness to serve on my committee. Their knowledge and expertise in the area was extremely valuable and greatly enriched my dissertation work.

I am also grateful to several fellow doctoral students at Georgia Tech. Although there is not enough space to mention everyone that has made this journey memorable, there are a few people that I must mention. I would not have survived my beginning years of classes and numerous homework assignments without my persistent study group of Bruce Cox, Hannah Smalley, and Jackie Griffin. Next, I should thank Antonio Carbajal for being such a supportive friend and office mate over the years. And last but not least, I have to thank Jackie Griffin for being an amazing office mate over the past six years. I could not have made it through this process without her and the countless hours of encouragement and counseling over the years.

I am extremely thankful to my parents for always supporting me in all of my endeavors. Their support and encouragement has allowed me to pursue all of my dreams and never doubt my ability to accomplish them. I am honored to continue the legacy of education in our family by being the fourth generation to complete a Doctor of Philosophy. And most of all, I would like to thank my husband, Adam, for his unending support during this process. Without his love and encouragement, finishing this dissertation would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This dissertation addresses problems arising in freight routing and scheduling where full truckload (FTL) and less-than-truckload (LTL) carriers are used to serve transportation needs. Each of the problems investigated in this dissertation tries to optimize/maximize consolidation to decrease system transportation costs by (1) carefully choosing the timing and path of freight and/or (2) introducing consolidation points. Approaches are proposed that enable effective planning and operation of freight routing and scheduling for large-scale transportation networks.

Chapter 2 presents solution approaches for a shipper pickup and delivery planning problem faced by many large retailers to move freight from suppliers to distribution centers. Each shipment is moved either direct via a LTL carrier or possibly consolidated with other shipments and moved by one or two FTL routes. When using a FTL carrier, the shipper takes advantage of contracted lane rates that establish prices per mile for a truck operated between two locations that are significantly less than the comparable LTL price for shipping a full truckload. Consolidated FTL routes may each visit multiple shipment origins (supplier locations) and/or destinations (distribution center locations). Additionally, FTL routes may move shipments through a single crossdock facility en route. The challenge in this planning problem is to exploit as much as possible negotiated truckload lane rates and to judiciously make use of routes through crossdock facilities to consolidate shipments. The primary contributions of this section are that (1) an interesting new problem variant is introduced to the field of transportation and logistics that is important in practice and (2) the solution approach demonstrates that exploiting knowledge of the problem and solution structure to cleverly select subsets of path variables for evaluation during each iteration of an integer programming based local search heuristic is effective on path-based routing models.

Chapter 3 evaluates how to route each customer shipment through a sequence of transfer terminals in a LTL carrier network. At each terminal stop, a shipment is unloaded from an inbound trailer and reloaded onto an outbound trailer. A load plan determines the specific sequence of terminal transfers to be used for freight moving between each origin and destination. The design of the load plan determines the linehaul transportation and handling costs required to serve customers. We develop an improved very large-scale neighborhood search heuristic for solving an integer programming model for load plan design. The main contributions of this section include (1) the investigation of the pros and cons of optimizing system-wide into a single destination versus optimizing freight for all destinations in a small region, and (2) a solution approach that can find load plans with costs 6 to 7% lower than those used in practice, and can find 2.5 to 5% additional cost savings using the same time budget when compared to an approach optimizing system-wide into a single destination.

Chapter 4 addresses a strategic planning problem that extends the load plan design problem to consider terminal roles. We investigate two-stage approaches that first identify the set of transfer terminals and then develop the corresponding load plan. Computational results compare the terminals chosen as transfer facilities from the proposed integer programming based local search method with a traditional hub location formulation and a simple facility location formulation to depict the benefits gained from modeling additional information. The key contributions of this section are (1) the introduction of a new hub location problem variant incorporating freight dispatch timing and trailer transportation cost characteristics found in the LTL trucking industry and (2) a solution approach utilizing IP-based local search that demonstrates the importance of incorporating freight dispatch timing.

Finally, Chapter 5 summarizes the main conclusions from this dissertation and discusses directions for further research.

CHAPTER I

INTRODUCTION

In the United States, nearly 70% of all freight transported annually is served by truck transportation, accounting for $671 billion worth of manufactured and retail goods [48]. The total revenue of the trucking industry is estimated to be about $650 billion and represents about 5% of U.S. Gross Domestic Product [5]. Because of the growing need and importance of transportation services, it should be no surprise that the trucking industry is projected to have an expected tonnage increase of 30% and revenue increase of 72% by 2018 [5]. The trucking industry is comprised of about 500,000 carriers, so the competition for the growing tonnage and revenue is very competitive. Therefore, in order for a single carrier to profit from these projected increases, cost-effective and competitive operations are a necessity. For this reason, the focus of this dissertation is on developing efficient freight routing and scheduling techniques for problems arising in large-scale freight transportation networks.

The freight service options provided by carriers are generally broken down into two categories, full truckload (FTL) or containerload services and less-than-truckload (LTL) services. FTL freight carriers offer direct service between an origin and destination pair for shipments between 10,000 and 50,000 pounds. In some cases, a FTL carrier allows a contracted truckload move to perform multiple-stop routes that pick up shipments from one or more pick-up locations and deliver them to one or more delivery locations for an additional cost. To discourage excessive extra travel distance in this case, the carrier may stipulate an upper limit on out-of-route travel either as an absolute bound or as a ratio with respect to the original distance between the origin and final destination of the multiple-stop route. On the other hand, LTL freight carriers provide service for shipments between 150 and 10,000 pounds. Because the shipments transported by LTL carriers typically account for only 5-10% of trailer capacity, direct shipment for each load between origin and destination is unrealistic. Therefore, LTL carriers operate linehaul networks made up of consolidation terminals that allow them to group multiple shipments from various origins to increase trailer utilization. The LTL linehaul network is the network of terminals and the transportation lanes connecting them, which commonly exhibits a hub-and-spoke topology for larger networks.

1.1 Freight Consolidation

There are opportunities to use both FTL and LTL service providers in transportation networks utilizing consolidation terminals, also referred to as consolidation networks. This dissertation addresses problems arising in freight routing and scheduling where FTL or LTL carriers are used to serve transportation needs. All of the work in this dissertation contains the common theme of optimizing/maximizing consolidation to decrease system transportation costs. There are two ways to accomplish this goal: (1) carefully choosing the timing and path of freight, and (2) introducing consolidation points. The problems investigated in this dissertation have one or both of these aspects. Approaches are proposed that enable effective planning and operation of freight routing and scheduling for large-scale transportation networks.

The problems presented here address strategic, tactical, and operational planning problems in two settings. The first setting investigates the use of contracted transportation options using a private consolidation network. The problems considered in this setting focus on developing models and approaches addressing an operational planning problem for a large retail company. The proposed techniques allow routing plans to be dynamically generated given a set of deterministic origin-destination demands for a given planning period. In the second setting, the focus is on the freight consolidation operations for a LTL carrier. The freight consolidation problems faced by a LTL carrier involve strategic and tactical decision making and are addressed using deterministic time-expanded network-based optimization models.

The first problem setting presents an operational planning problem faced by large retail chains offering products across various market segments, such as grocery, home improvement, and clothing, that must transport goods from many suppliers to many retail outlets.

To enable low store inventories, outlets are served with frequent shipments from distribution centers, which may span the spectrum from the traditional warehouses to crossdocks. Major retailers then have two major freight transportation subproblems: one concerned with moving product from suppliers to distribution centers, and the other with moving product from distribution centers to stores. This dissertation focuses on models for building cost-effective inbound transportation plans into distribution centers.

When individual distribution centers maintain high inventory levels of a product, or when large suppliers deliver many products to a distribution center simultaneously, inbound transportation may be relatively simple. FTL services direct from supplier to DC are economical choices in these cases. In other cases, the simplest options are more expensive. For example, public LTL freight services can be used to move smaller quantities direct, but the rates for such shipments (per weight and per mile) are significantly higher than FTL service. In this context, large shipments are greater than 10,000 pounds and typically served by FTL service, where as smaller shipments are less than 10,000 pounds and are typically served by LTL service options. We consider a more complex inbound system that has (1) multi-stop FTL routes and (2) crossdock options so that the cost economies of FTL shipping can be exploited.

In the second problem setting, we first address a tactical planning problem that determines the freight paths to maximize consolidation for a LTL carrier. LTL carriers route each customer shipment through a sequence of terminal stops en route from origin to destination using a consolidation network. At specified terminal stops, a shipment is unloaded from an inbound trailer and reloaded onto an outbound trailer. A *load plan* for large LTL carriers determines the specific sequence of terminal stops to be used to transfer freight between each origin and destination. The design of the load plan determines the linehaul transportation and handling costs required to serve customers. Effective load plans are designed to minimize total linehaul transportation and handling costs, while satisfying origin-to-destination maximum transit time requirements for customers.

Secondly, we investigate a strategic planning problem that considers which subset of terminals in a LTL network should be operated as transfer facilities. This subset of transfer facilities, or breakbulks (BBs) as they are called in the LTL trucking industry, are operated as cross-docking facilities, where freight is transferred between vehicles to increase vehicle utilization. The set of chosen BBs directly impacts the freight routing operations and the resulting load plan. Therefore, the impact of freight dispatch timing and trailer transportation costs on the choice of BB terminals is examined.

1.2 Integer Programming Based Local Search

A second theme throughout the work presented in this dissertation is the use of integer programming (IP) based local search as an alternative to solving large detailed optimization models or using metaheuristics. Due to the size of problems seen in practice, large detailed models often become intractable and/or impossible to solve in a limited time period using commercial solvers. Therefore, solution approaches are developed in this work that utilize a combination of heuristic search and optimization to address problems that are too large for direct optimization techniques. Solution approaches combining exact optimization and heuristic search techniques have been proposed for many different problems, including the classic network design (e.g. see [25]), vehicle routing (e.g. see [18], [3], [22]), inventory routing (e.g. see [45]), covering salesman (e.g. see [44]), maximal covering (e.g. see [26]), and multidimensional knapsack problem (e.g. see [27]). In the framework used in this work, the local search methodology evaluates neighborhoods solving IP formulations smaller than the original detailed model. These IP formulations may be a simple restriction of the detailed model formulation or a tailored formulation that can quickly evaluate the given neighborhood to identify improvements. In Chapter 2, the IP-based neighborhood search methodology presented uses a restricted version of the path-based IP model to evaluate restricted subsets of paths in varying neighborhoods. In Chapter 3, a tailored IP formulation was developed to evaluate neighborhoods during the local search routine. This tailored IP formulation allows a larger neighborhood to be evaluated more efficiently than can be evaluated using a restricted version of the detailed model for the same neighborhood. And in Chapter 4, the IP-based neighborhood search methodology presented uses both restricted versions of the detailed model formulation and tailored IP formulations to evaluate neighborhoods and find improving solutions.

The general framework of the IP-based local search approaches developed in this dissertation indicate that this solution technique may be successful in other transportation and logistics problem settings. The characteristics of the formulations used to search the neighborhoods are the same as those found in many transportation settings. These characteristics include trailer based transportation costs, integer counting constraints to determine the number of trailers, and assignment constraints choosing a single variable used to determine a single path assignment from a set of options.

1.3 Outline and Contributions

We conclude this introductory chapter by outlining the remainder of this dissertation, and describing the contributions of each of the following chapters.

Chapter 2 presents solution approaches for a shipper pickup and delivery planning problem to move freight from suppliers to distribution centers. Each shipment is moved either direct via a LTL carrier, or possibly consolidated with other shipments and moved by one or two FTL routes (one into a crossdock and one out of a crossdock). When using a FTL carrier, the shipper takes advantage of contracted lane rates that establish a price per mile for a truck operated between two locations that are significantly less than the comparable LTL price for shipping a full truckload. The challenge for the shipper is to consolidate multiple shipments effectively to take advantage of this price differential. Consolidated FTL routes may visit multiple shipment supplier locations and/or distribution center locations. Additionally, FTL routes may move shipments through a single crossdock facility en route. A path-based IP model for this planning problem is presented. The model can be solved directly with commercial integer programming software for smaller instances. For larger instances, an IP-based local search approach is developed. A computational study using data from a major U.S. retailer demonstrates the effectiveness of the solution approaches. In each instance, substantial transportation cost savings are identified from baseline LTL costs. The main contributions of this chapter are that (1) an interesting new problem variant is introduced to the field of transportation and logistics that is important in practice and (2) the solution approach demonstrates that exploiting knowledge of the problem and solution structure to cleverly select subsets of path variables for evaluation during each iteration of an IP-based local search heuristic is effective on path-based routing models.

Chapter 3 develops and tests a new heuristic solution method for solving a load plan design optimization problem on a large-scale consolidation network used by a LTL carrier, building on prior work in Erera et al.[19]. In that paper, the authors propose a detailed time-space network representation of freight routing and consolidation decisions that more accurately models the trailer flows that result from a given load plan given modern LTL operations. Erera et al. [19] introduce an IP-based local search heuristic solution approach that focuses on system-wide optimization for freight into a single destination. In this chapter, we propose an approach that focuses on localized optimization of all freight in a region. To speed up the solution time for individual iterations (which in turn allows more heuristic iterations within a time budget), we also experiment with a new approach for handling empty trailer movements where we remove trailer balance constraints (and associated variables) from the neighborhood IPs and replace with lower bounds. Computational results using instances from a large US LTL carrier show that the approach can find load plans with substantial cost improvements generated by increased freight consolidation accounting for over $300,000 per week. The main contributions of this chapter include (1) the investigation of the pros and cons of optimizing system-wide into a single destination versus optimizing freight for all destinations in a small region, and (2) a solution approach that can find load plans with costs 6 to 7% lower than those used in practice, and can find 2.5 to 5% additional cost savings when compared to the original approach using the same time budget.

Chapter 4 presents a solution approach used to identify the set of transfer facilities operated in a LTL transportation network. This chapter addresses a strategic planning problem that extends the load plan design problem to consider terminal roles, thus identifying the set of transfer terminals and corresponding load plan. The load plan neighborhood search methodology from the previous chapter is integrated into an IP-based local search approach that also uses restrictions of the full IP to help focus the search on identifying the best terminals to operate as transfer facilities. Computational results show that modeling trailer transportation cost and freight dispatch timing information enables better transfer facilities to be chosen when compared with traditional hub location techniques, resulting in lower total transportation and handling costs in the corresponding load plan. The main contributions of this chapter include (1) the introduction of a new hub location problem variant incorporating freight dispatch timing and trailer transportation cost characteristics found in the LTL trucking industry and (2) a solution approach utilizing IP-based local search that demonstrates the importance of incorporating freight dispatch timing.

Finally, Chaper 5 concludes this dissertation by discussing potential extensions for each of the problems addressed.

CHAPTER II

A PICKUP AND DELIVERY PROBLEM USING CROSSDOCKS AND TRUCKLOAD LANE RATES

2.1 Introduction

The U.S. retail industry continues to evolve toward a model where a few large and powerful players sell a growing number of products. In this setting, large retail chains offering products across various market segments, such as grocery, home improvement, and clothing, must transport goods from many suppliers to many retail outlets. To enable low store inventories, outlets are served with frequent shipments from distribution centers, which may span the spectrum from traditional warehouses to cross-docking facilities (crossdocks). Major retailers then have two major freight transportation subproblems: one concerned with moving product from suppliers to distribution centers, and the other with moving product from distribution centers to stores. This research focuses on models for building cost-effective transportation plans inbound to distribution centers.

When individual distribution centers maintain high inventory levels of a product, or when large suppliers deliver many products to a distribution center (DC) simultaneously, inbound transportation may be relatively simple. Full truckload (FTL) or containerload services direct from supplier to DC are economical choices in these cases. In other cases, the simplest options are more expensive. For example, public less-than-truckload (LTL) freight services can be used to move smaller quantities direct, but the rates for such shipments (per weight and per mile) are significantly higher than FTL service.

In this research, we consider a more complex inbound system organized to utilize contracted FTL services inbound to DCs for smaller shipments. To do so, the retailer no longer uses direct supplier-to-DC shipments exclusively. Instead, multiple-stop FTL routes that pick up shipments from one or more suppliers and deliver them to one or more DCs will also be considered. In addition, the retailer may operate additional inbound crossdocks that

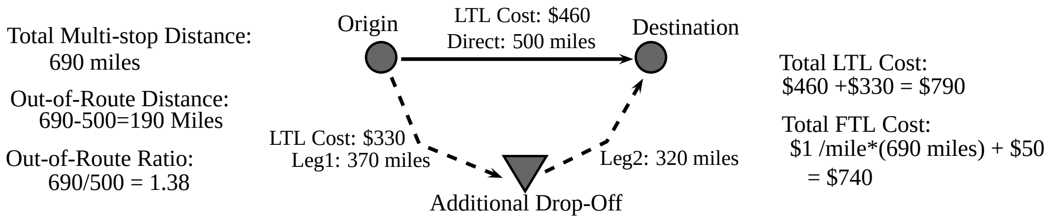


Figure 1: Lane Rate Additional Stops

allow full truckloads to be sent from suppliers to crossdocks, and then from crossdocks to DCs.

When FTL services are used for multiple-stop routes, determining transportation charges is more complex. Retailers negotiate contracts with FTL carriers that specify a transportation charge (or lane rate) for a large number of frequently used origin-destination pairs, or lanes. Each lane rate represents a price per mile for a truck operated on this lane, and is significantly less than the comparable price per mile of shipping a full truckload of LTL freight. In addition to the lane rate, the contract with the FTL carrier will often also allow additional stops between the lane origin and destination for picking up or dropping off freight at additional cost. To discourage excessive extra travel distance in this case, the carrier may specify an upper limit on out-of-route travel either as an absolute bound or as ratio with respect to the original lane distance. If this bound is not violated, the FTL company simply charges the retailer the lane rate multiplied by the total route travel distance including the added stops, along with appropriate stopoff charges. Figure 2.1 gives an example of adding an additional stop to a lane with a maximum out-of-route travel distance ratio allowance of 1.40, a truckload rate of $1 per mile, and an additional stop charge of $50. The example shows that a single truckload route traveling on path p = (Origin, Additional Drop-Off, Destination) can replace two LTL shipments traveling between (Origin, Destination) and (Origin, Additional Drop-Off), with total cost savings of $50.

This chapter focuses on a pickup and delivery planning problem to move shipments from suppliers to DCs in this setting; see Savelsbergh [46] for a survey of the pickup and delivery problem literature. A retailer needs to plan inbound moves serving a set of shipments, where each shipment is served either by an LTL route or by one or two FTL routes. FTL routes may include additional stops between origin and destination, and may originate from or terminate at a crossdock; LTL routes directly connect a shipment's origin to destination. Note that since FTL routes in this context are not cycles, this problem can be considered a type of *open vehicle routing* problem. LTL lane rates exist between each shipment's origin and destination, while FTL rates exist for a subset of network lanes. The problem is to find a feasible set of open routes that minimizes total transportation cost, defined as the sum of the minimum transportation cost for each route in the set. A feasible set of open routes is such that: (1) every shipment is served, traveling via either a single route from origin to destination or a sequence of one route from the origin to a crossdock and one route from a crossdock to the destination; and (2) the shipments simultaneously loaded into any vehicle have total weight and cubic volume (cube) that do not exceed vehicle limits.

We model the problem using a path-based integer programming (IP) formulation, and develop solution approaches tailored to solve problem instances representative of those found in practice using data from a large U.S. retailer. Solving the IP model directly using commercial integer programming software works well on instances with a smaller number of shipments, quickly finding an optimal solution. However, for larger instances, solving the IP model becomes more difficult. In practice, routing decisions need to be made in a restricted period of time; our industry partner suggests that they allocate no more than four hours of computation time. Since solvers struggle to close the optimality gap for larger instances given this time budget, we also develop a heuristic search scheme in which restrictions to the integer program are solved sequentially. Computational experiments a number of instances based on data from a large U.S. retailer compare the performance of both the direct optimization and heuristic approaches. Experimental results demonstrate that the heuristic approach provides high quality solutions for all instances, given the time restriction. Furthermore, in all instances, substantial cost savings are identified using the model when compared against the baseline LTL costs.

The remainder of the chapter is organized as follows. Section 2.2 provides a brief literature review of related research. Sections 2.3, 2.4, and 2.5 formally define the problem and detail the proposed solution approaches. Section 2.6 provides computational results from a real world instance.

2.2 Literature Review

The primary focus herein is to develop approaches for planning transportation for origin-destination less-than-truckload shipments to take advantage of cost economies in truckload transportation and the consolidation opportunities provided by shipping through cross-docks. The existing literature in this area can be grouped roughly into three categories: (1) cross-docking network planning; (2) vehicle routing with cross-docking; and (3) pickup and delivery routing with transfers or transshipments.

Cross-docking network planning problems are primarily service network design problems, and the existing literature does not consider multiple-stop vehicle routing decisions. One example in this category is Chen *et al.* [11], who investigated planning shipments with time windows through a network of crossdocks with handling capacities and different handling costs. They showed that the most cost-effective solutions do not always send each shipment through the crossdock that minimizes out-of-route distance (and transportation cost), due to capacities and handling costs. Unlike our research, however, their paper assumed that total transportation cost can be modeled as a linear function of individual shipment costs (similar to LTL costs) and did not explicitly model consolidation in vehicles. Li *et al.* [34] addressed a crossdock routing problem for a single product type, where supplies of the product from origins need to be routed to customer demand locations. Suppliers and customers both have time windows defining when shipments must be shipped and received. Inventory may be held at crossdocks, but creates holding cost. Although their paper only considered direct shipments (origin-to-destination, origin-to-crossdock, crossdock-to-destination), constraints were used to count the number of required capacitated vehicles on each transport arc and using an additional vehicle incurs a fixed charge. Sung and Song [47] addressed a similar problem with two types of vehicles with different operating costs and capacities. While they did not model time windows for pickup or delivery, they enforced a service time limit on the maximum duration (travel plus handling time) for each shipment. Furthermore, the formulation included variables for the selection of which potential crossdocks to open to serve demand. Jayaraman and Ross [30] also considered a design problem of which crossdock facilities to open to serve demands for a set of products. Like Li *et al.* [34], the model decided where to source demand for each product for each customer rather than planning for known shipments. They developed a two phase modeling approach, where the first phase model addressed the design of the network and determines fixed supplier-to-crossdock and crossdock-to-customer assignments. In the second phase, all supplier and crossdock location decisions were fixed, and more flexible product routing is allowed. Transportation costs were assumed to be linear in product flow.

Prior papers have addressed vehicle routing problems where a crossdock is used to serve demand from multiple origins moving to multiple destinations. The specific planning problem considered is often a variation of the so-called *vehicle routing problem with cross-docking* (VRPCD). In this problem, a homogeneous fleet of vehicles based at a crossdock performs a set of pickup tours, bringing shipments from origins into the crossdock. Then, the same fleet performs a set of delivery tours from the crossdock to destinations. All pickups and deliveries must occur within specified time windows, and shipment timing coordination is enforced at the crossdock. Lee *et al.* [33] proposed a mixed integer programming formulation and a tabu search algorithm, assuming that all vehicles should simultaneously arrive back at the crossdock after pickup tours. Liao *et al.* [35] developed a new tabu search algorithm with minor differences from the approach in [33] that outperformed the original algorithm on test problems, finding both lower cost solutions and requiring less computation time. Wen *et al.* [51] addressed a similar problem, but considered a more flexible coordination constraint at the crossdock that guaranteed that outbound tours cannot be dispatched until all of their freight has arrived inbound to the crossdock. A tabu search heuristic with an embedded adaptive memory procedure found cost-minimizing routes.

Many-to-many pickup and delivery problems that model vehicle consolidation, as extensions of one-to-many or many-to-one multiple vehicle routing problems, have received significant attention in the literature. Pickup and delivery problems with opportunities for freight or passenger transfer between vehicles have been studied less, but have not been ignored. For example, in dial-a-ride passenger transportation one research example is provided by Cortes *et al.* [13]. The authors extended the classical pickup and delivery problem with an option for passengers to transfer from one vehicle to another at specific locations. A Benders decomposition approach for solving the resultant formulation was proposed, and compared to a straightforward branch-and-bound strategy.

In freight settings, researchers have proposed transshipment extensions to the pickup and delivery problem with time windows. A notable example is the work of Mitrovic-Minic and Laporte [36]. Their paper presented a two-phase heuristic for the problem, and used the approach in an empirical study based on a courier company serving a large geographic region. By utilizing transshipments at fixed locations, loads could be served by two vehicles in sequence. One vehicle picks up the load and takes it to the transshipment location, where the load is transferred to the second vehicle that transports it to the delivery location. Examples with a relatively small number of loads and transshipment points were shown to gain significant benefit from transshipment when origins and destinations were clustered. Mues and Pickl [37] considered a similar problem, where different transportation modes were available. A column generation approach was proposed for two versions of the problem, one with a single transshipment location and another with multiple transshipment locations. Results were provided for the single location version, and instances with up to 70 loads could be solved quickly.

The research we present in this chapter is similar to the cross-docking network planning papers. However, we assume that the available crossdocks are known, and furthermore that known shipments with fixed origins and destinations must be served. Since the shipment quantities are small, we allow planning multiple stop pickup-and-delivery routes to serve these shipments using the best possible truckload rates, potentially using a single crossdock en route as a transfer point. To our knowledge, no existing research presented in the literature has focused on this important planning problem.

2.3 Problem Definition and Proposed Solution Approach

We now define a pickup and delivery open routing problem with crossdocks where FTL rates can be used on a subset of lanes. Let K be a set of orders that must be moved within a planning horizon. Each order $k \in K$ is to be moved from an origin location o_k to a destination location d_k, and has a weight of $w_k \leq W$ and cubic volume of $q_k \leq Q$ where W and Q are the weight and cube capacity of a truckload. Let J be a set of crossdocks. Assume that a LTL shipping rate of $r^L(o_k, d_k)$ per pound per mile is available for all $k \in K$. Assume that a FTL shipping rate of $r^T(i,j)$ per mile, an extra stop charge of $s^T(i,j)$, and a truckload minimum cost $r^T_{\min}(i,j)$ exists for each (i,j) in some subset A^T of all lanes $V \times V$, where $V = J \cup \{\cup_{k \in K} o_k\} \cup \{\cup_{k \in K} d_k\}$. Let M be a maximum out-of-route distance ratio allowed for truckload moves. Let $\ell(p)$ be a known function that returns the travel distance required to travel the sequence of locations p. The problem is then to determine a set of a feasible routes R^* where:

1. each order $k \in K$ is transported from o_k to d_k via a single route, or via two routes connecting at a single crossdock $j \in J$;

2. each route $p = \{o_{(1)}, o_{(2)}, ..., o_{(a)}, j\} \in R^*$ destined to a crossdock j visits only order origins, and must not exceed vehicle capacity: $\sum_{i=1}^{a} w_{(i)} \leq W$ and $\sum_{i=1}^{a} q_{(i)} \leq Q$;

3. each route $p = \{j, d_{(b)}, d_{(b-1)}, ..., d_{(1)}\} \in R^*$ from a crossdock j only visits order destinations, and must not exceed vehicle capacity: $\sum_{i=1}^{b} w_{(i)} \leq W$ and $\sum_{i=1}^{b} q_{(i)} \leq Q$;

4. each route $p = \{o_{(1)}, ..., o_{(a-1)}, o_{(a)}, d_{(a)}, d_{(a-1)}, ..., d_{(1)}\} \in R^*$ that does not visit a crossdock will serve one or more orders, visiting all origin locations followed by all destination locations in last-in first-out order, and must not exceed vehicle capacity: $\sum_{i=1}^{a} w_{(i)} \leq W$ and $\sum_{i=1}^{a} q_{(i)} \leq Q$;

5. each route $p_O(j) \in R^*$ outbound from a crossdock j can only serve destinations of orders whose origins are also served on some inbound route $p_I(j) \in R^*$; and

6. the sum $\sum_{p \in R^*} c(p)$ of the individual route transportation costs is minimized.

Each route cost $c(p)$ is defined to be the minimum cost visiting locations $p = \{v_1, v_2, ..., v_m\}$ given r^L, r^T, s^T, and M; we will show that $c(p)$ can be determined for any p by solving an acyclic minimum cost path problem.

2.3.1 Path-Based Formulation

We propose solution approaches that rely on path-based binary integer programming models. Each path in the model represents a route of one of the three types: inbound from order origins to crossdock, outbound from crossdock to order destinations, or pickup visits at origins followed by delivery visits at destinations for a set of orders. Let R_I be all capacity-feasible routes inbound to all crossdocks, R_O be all capacity-feasible routes outbound from all crossdocks, and R_D be all capacity-feasible last-in first-out pickup-and-delivery routes. Let $R = R_I \cup R_O \cup R_D$. Furthermore, for a specific order k let $R_I(k, j)$ be the subset of routes in R_I that pickup order k from o_k and deliver it to crossdock j, $R_O(k, j)$ be the subset of routes in R_O that depart crossdock j with order k and deliver it to d_k, and $R_D(k)$ be the subset of routes in R_D that serve order k by visiting o_k and d_k.

The decision variables are then:

$$x_p = \begin{cases} 1 & \text{if route } p \text{ is selected} \\ 0 & \text{otherwise} \end{cases},$$

and the integer programming model is:

$$\begin{aligned}
\text{minimize} \quad & \sum_{p \in R} c(p) x_p && && (1)\\
\text{subject to:} \quad \sum_{p \in R_D(k)} x_p + \sum_{j \in J} \sum_{p \in R_I(k,j)} x_p &= 1 && \forall k \in K && (2)\\
\sum_{p \in R_O(k,j)} x_p &= \sum_{p \in R_I(k,j)} x_p && \forall k \in K, j \in J && (3)\\
x_p &\in \{0, 1\} && \forall p \in R && (4)
\end{aligned}$$

The objective function (1) minimizes the total transportation costs by summing the costs of chosen routes. Constraints (2) ensure that each individual order k is served by exactly one pickup route. Constraints (3) further ensure that each order k that is delivered into crossdock j must also be served on an outbound route from j delivering to d_k.

2.3.2 Calculating Route Costs

The cost to fulfill an order depends on how it is shipped from its origin to its destination. The default, most costly, option is to dispatch it as an LTL shipment, which is modeled as a single-order route $p \in R_D$ from the supplier associated with the order to the distribution center associated with the order. This option is available for all orders.

The goal of the optimization is to identify cheaper cost options for some or all orders by exploiting less costly FTL lane rates, either for an individual order or by combining orders into multi-stop routes (either direct multi-stop routes or indirect multi-stop routes visiting a crossdock).

If a route $p \in R_D$ serves a single order k, its cost is determined by the minimum cost prescribed by either the LTL rate or the FTL rate (if it exists) on lane $p = (o_k, d_k)$, as follows:

$$c(p) = \min \left\{ r^L(p) * w_k * \ell(p), \max\{r^T(p) * \ell(p), r^T_{\min}(p)\} \right\}, \tag{5}$$

where $r^T_{\min}(p)$ is the minimum truckload charge for lane p.

Determining the cost of a multi-stop route $p \in R$ is more involved. In order for such a route to be feasible, it must be covered by the set of available truckload lane rates. We have the following definition.

Definition 2.3.1 (Covered Route) *A route $p = \{i_1, i_2, ..., i_m\} \in R$ is covered if there exists a rated path p_r from i_1 to i_m,* i.e.*, a subset of the stops in p (in sequence) such that each $(i_j, i_{j+k}) \in p_r$ has an associated truckload lane rate $r^T(i_j, i_{j+k})$ and the subsequence $\{i_j, ..., i_{j+k}\}$ of p does not exceed the maximum out-of-route distance ratio: $\frac{\ell(i_j, i_{j+1}, ..., i_{j+k})}{\ell(i_j, i_{j+k})} \leq M$.*

A simple example of a covered route p would be one where a truckload rate exists on lane (i_1, i_m), and $\frac{\ell(p)}{\ell(i_1, i_m)} \leq M$. In this case, the cost of route p could be determined by $\max\{r^T(i_1, i_m)\ell(p) + (m-2)s^T(i_1, i_m), r^T_{\min}(i_1, i_m)\}$. Note that for each of the $m - 2$

additional stops visited between i_1 and i_m, the lane stop-off charge is incurred. It is possible, however that p might be also covered by an alternative rated path p_r, and using that path may result in a lower cost $c(p)$.

To determine the minimum cost $c(p)$ of any covered route $p \in R$, we use a simple acyclic shortest path problem. A network is constructed with m nodes, labeled $\{i_1, i_2, ..., i_m\}$. An arc connects i_j to i_{j+k} for $k \geq 1$ if and only if a truckload lane rate $r^T(i_j, i_{j+k})$ exists and if using this lane rate would not violate the out-of-route distance ratio: $\frac{\ell(i_j, i_{j+1}, ..., i_{j+k})}{\ell(i_j, i_{j+k})} \leq M$. The cost of such an arc is then given by:

$$c_{i_j, i_{j+k}} = \max\{r^T(i_j, i_{j+k})\ell(i_j, i_{j+1}, ..., i_{j+k}) + (k-1)s^T(i_j, i_{j+k}), r^T_{\min}(i_j, i_{j+k})\}$$

The cost $c(p)$ is determined by the cost of a minimum cost path connecting i_1 to i_m on this acyclic network. Note that the existence of at least one such path is guaranteed, since p is a covered route.

2.4 Building a Route Set

In practice, enumerating the full set of feasible routes R will likely lead to computationally intractable instances of the IP model. Therefore, we develop a pragmatic solution approach that enumerates a large subset of feasible routes $R' \subset R$ that are likely to be part of a high-quality solution.

Given a set of orders $K(p)$ to be served on route p, consider the problem of determining the best sequence of stops. There are three types of routes: (1) direct routes that visit both o_k and d_k for each $k \in K(p)$; (2) inbound crossdock routes that visit o_k for each $k \in K(p)$ before terminating at some crossdock $j \in J$; and (3) outbound crossdock routes that depart crossdock $j \in J$ and then visit d_k for each $k \in K(p)$. For each type of route, a sequential insertion approach is used to construct route p by inserting one order k at a time.

For a direct route of type (1), consider a partial route p' visiting a subset of the locations required for the orders in $K(p)$, *i.e.*, $p' = \{o_1, o_2, ..., o_m, d_m, d_{m-1}, ..., d_1\}$. When inserting order k into p', locations o_k and d_k are inserted simultaneously with o_k immediately prior to o_1 or immediately after $o_1, o_2, ..., o_m$, which then fixes the position of d_k immediately after d_1 or immediately prior to $d_1, d_2, ..., d_m$ to maintain the LIFO ordering. For routes of type

(2) into crossdock j, a partial route will have the form $p' = \{o_1, o_2, ..., o_m, j\}$ and inserting order k requires finding an insertion position for o_k immediately prior to any location in p'. Similarly, for routes of type (3) from crossdock j, inserting order k into $p' = \{j, d_1, d_2, ..., d_m\}$ requires finding an insertion position for d_k immediately after any location in p'.

Given an order k and a partial path p', all feasible insertion positions are first evaluated to find a minimum cost feasible insertion position. An insertion position is considered feasible if the new route p'' after the insertion of order k is covered. For each such feasible position, the new route cost $c(p'')$ is determined using the approach in Section 2.3.2, and the cheapest position is selected. Note that if path p'' visits the same location multiple times in sequence (for example, if two or more orders are to be picked up at the same supplier location), these visits are merged into a single visit to the location before costing. There may not exist any feasible insertion position for order k in p'. If order k is not the last order to be added to p', we insert it at the position that minimizes the increase in total route distance from p' to p'' in the hope that a covered route will result after inserting additional orders. Each covered final route p is included in R' with cost $c(p)$.

Computing $c(p'')$ exactly for each insertion position is somewhat more expensive computationally than traditional insertion cost calculations. However, computational experiments (see Section 3.6 for details) demonstrate the benefit of this strategy over a simpler one where best insertion positions are determined by minimizing the increase in the total distance of p'' over p'.

Next, we describe the strategies used to generate sets of orders to be served together on a route. To ensure that a feasible solution to the IP exists, we first generate a single-order direct route for each $k \in K$ of the form $p = (o_k, d_k)$, with a cost $c(p)$ given by (5).

To generate sets of orders to be served together on a route, a two-step procedure is used. In the first step, we create order lists that contain a reasonably large number of orders. In the second step, each of the order lists is partitioned into sets of orders to be served together on a route using a first-fit bin packing procedure. The bin-packing procedure works as follows. Let O be the order list. Orders are removed from O in sequence and placed into bins, where bin i contains the orders $K(p_i)$ for route p_i. Order $k \in O$ is placed in the first available bin

p_i with $w_k \leq W - \sum_{k' \in K(p_i)} w_{k'}$ and $q_k \leq Q - \sum_{k' \in K(p_i)} q_{k'}$.

To create the order lists, we group orders by the geographic proximity of order origins and destinations. Let $L(S)$ and $L(D)$ be the sets of physical locations representing suppliers (order origins) and distribution centers (order destinations) respectively. Then, given a distance ρ, let the neighbor list $N(f, \rho)$ for each facility $f \in L(S) \cup L(D)$ be the facilities $f' \in L(S) \cup L(D)$ no further than ρ miles from f. By only generating routes that serve orders whose facilities are nearby geographically, we hope to generate low cost routes. See Section 3.6 for a brief discussion on how ρ was chosen in this study.

Order lists for direct routes. We start by creating two order lists $O_1(i)$ and $O_2(i)$ for each location $i \in L(S) \cup L(D)$. If i is a supplier, $O_1(i)$ and $O_2(i)$ contain the orders $k \in K$ with $o_k = i$. The difference between the order lists is the way in which the orders are sorted. In $O_1(i)$ the orders are sorted in order of non-increasing weight w_k. In $O_2(i)$ the orders are sorted in order of non-increasing angle θ_k of the destination location d_k when the location is represented in polar coordinates. (When there are orders with the same destination location, the orders are further sorted in non-increasing order of weight w_k.) Next, we create two order lists for pairs of suppliers i and i' such that $i' \in N(i, \rho)$ (and thus $i \in N(i', \rho)$). Each of these two lists contains the orders $k \in K$ with $o_k = i$ or $o_k = i'$. Again, $O_1(i, i')$ is sorted in non-increasing order of weight w_k and $O_2(i, i')$ is sorted first by the angle θ_k of destination d_k and then by non-increasing order by weight w_k. The process is repeated for distribution centers. The creation of lists focused on order weights and lists focused on geographic locations provides diversification.

Order lists for crossdock routes. We limit τ, the number of crossdocks, that we consider for a particular order. Therefore, we start by creating crossdock order lists $K_j(\tau)$ for $j \in J$. For each order $k \in K$, we compute the detour travel distance incurred when order k is routed through crossdock $j \in J$, *i.e.*, $\ell(o_k, j, d_k) - \ell(o_k, d_k)$. Order k is then added to crossdock order list $K_j(\tau)$ for the τ lowest detour travel distance crossdocks j.

We use the crossdock order lists to construct the order lists that will be input to the bin packing procedure. For each crossdock $j \in J$, we first build inbound order lists $O_1(i) \subseteq$

$K_j(\tau)$ for each unique supplier location i. As before, the orders in $O_1(i)$ are sorted in order of non-increasing weight w_k. Then, we build multiple-supplier inbound order lists by adding the orders in $K_j(\tau)$ for additional suppliers $i' \in N(i, \rho)$ one supplier at a time. For example, lists $O_1(i, i')$ and $O_1(i, i', i'')$ would be created if both i' and i'' are in $N(i, \rho)$. Orders for additional suppliers are added until a maximum number of suppliers is reached.

A similar process is used to create outbound order lists from each $j \in J$ to distribution centers. An outbound order list $O_1(i) \subseteq K_j(\tau)$ is created for each unique distribution center location i, and then additional order lists are created by adding orders bound for neighboring distribution centers $i' \in N(i, \rho)$.

2.5 Search with Restricted Route Sets

As instances become larger, *i.e.*, as the numbers of orders and possible routes grow, solving the integer program defined in Section 2.3.1 becomes increasingly difficult. To ensure that high-quality solutions are found reliably given limited available computation time, we have developed a search scheme that solves a sequence of smaller integer programs, each obtained by restricting the set of variables considered. This approach ensures that we maintain the two essential characteristics of our solution approach: (1) using heuristics focusing on local considerations to build routes, and (2) using an integer program focusing on global considerations to select the most appropriate routes. Search schemes based on restricting the set of variables considered in an integer program have become quite popular and have proven successful, see *e.g.*, De Franceschi *et al.* [21] and Hewitt *et al.* [24].

The restricted integer programs always include variables corresponding to the routes in the best known feasible solution. Thus, for each shipment, the integer program chooses between the route(s) that serve that shipment in the best known solution, and the alternative routes available to serve that shipment. Since even the restricted integer program may be difficult to solve, a computation time limit is imposed. The success of the search scheme depends on choosing restrictions R'' of the route set R' that result in improved feasible solutions quickly. (It is not important to prove optimality of an improved solution to the restricted problem solved.)

Our approach for restricting the route set R' is to focus on different types of problem locations each iteration. More specifically, we build the following restricted route sets R'':

1. **Distribution Center Restriction**

   Given a single distribution center $f \in L(D)$, let $K' \subset K$ be the set of orders that have f as the destination. Next, let $K'' \subset K$ be the set of orders for which there exists at least one route in R' that includes a visit to either o_k or d_k for an order $k \in K'$; the orders in K'' can be served on routes with orders in K'. The restricted set R'' contains all routes that serve *only* shipments in $K' \cup K''$.

2. **Crossdock Restriction**

   Given a crossdock $j \in J$, let $K' \subset K$ be the set of orders for which there exists at least one route that ends and one route that starts at j. The restricted set R'' contains all routes that end or start at j and serve only orders in K'.

3. **All Crossdocks Restriction**

   Let $K' \subseteq K$ be the set of orders for which there exists at least one route that ends and one route that starts at some crossdock $j \in J$. The restricted set R'' contains all routes that end and start at any crossdock and serve only orders in K'.

4. **No Crossdock Restriction**

   Let $K' \subseteq K$ be the set of orders for which there exists at least one route that does not include a visit to a crossdock. The restricted set R'' contains all routes that do not visit a crossdock and that serve only orders in K'.

5. **Suppliers Restriction**

   For this restriction, we consider the set of all orders K. The restricted set R'' contains all routes in R' that include a visit to any shipment origin o_k; the only routes not included here are those that begin at a crossdock.

The restricted route sets that we use represent a balance between relatively small ones, such as the Distribution Center Restriction, and relatively large ones, such as the All Crossdocks Restriction or the Suppliers Restriction. The overall search heuristic follows the scheme presented in Algorithm 1. The time limit T for the main part of the scheme is set to 3.5 hours. To ensure that with high probability each Distribution Center Restriction is solved at least once, the time limit for these restrictions is set based on the number of DCs; the maximum time allotted to each is $\frac{T}{|L(D)|}$. All other restricted IPs are allotted 10 minutes of solve time, with one exception. The final full IP is solved with a time limit of 20 minutes. It has proven valuable to include the solution of the full IP at the end, once a good solution and its corresponding upper bound is known, since it provides a global view of the problem and is sometimes able to find additional improvements.

Algorithm 1: IP-Based Neighborhood Search Scheme

```
Construct initial feasible solution;
while the search time has not exceeded time limit T do
    for every distribution center do
        Solve Distribution Center Restriction;
    end
    Solve Suppliers Restriction;
    Solve All Crossdocks Restriction;
    if All Crossdocks Restriction does not solve to optimality then
        for every crossdock do
            Solve Crossdock Restriction;
        end
    end
    Solve No Crossdock Restriction;
    Solve IP;
end
Solve No Crossdock Restriction;
Solve IP;
```

An initial feasible solution is found by first solving the linear relaxation of the full IP, fixing any route variable with a value less than 0.001 to zero, and then solving the resulting IP for ten minutes or until a feasible solution has been found; in practice, this approach always identifies a feasible solution within ten minutes of solve time.

2.6 Computational Results

The solution approach described above is implemented in C++ using ILOG Concert Technology with ILOG CPLEX 12.2 as the solver; when solving IPs all parameters are set to default values, except that multithreading was disabled. The experiments were run on a cluster of workstations with clockspeeds between 2.0 and 3.5 MHz and 1.5Gb of memory. Table 1 presents the characteristics for the instances used in our computational study. The first three instances are based on real-life data from a large U.S. retailer, and represent sets of shipments to be transported in an operating period of about a week. The remaining instances have been generated using the suppliers, distribution centers, crossdocks (CDs), shipments, and lanes with FTL rates from the first three instances.

Table 1: Instance Characteristics

Instance	# Suppliers	# Distribution Centers	# CDs	# Shipments	# FTL Lane Rates	Max $\Delta Cost$ Bound
Instance 1	4	40	14	128	1389	67.56%
Instance 2	98	46	14	507	7864	78.50%
Instance 3	33	40	14	1126	3073	78.63%
Instance 4	73	15	14	503	4082	77.89%
Instance 5	53	49	8	389	2678	81.78%
Instance 6	89	38	10	612	5772	80.56%
Instance 7	44	48	10	254	2736	72.92%
Instance 8	41	61	10	615	3914	75.83%
Instance 9	85	59	10	885	7270	75.97%
Instance 10	80	35	10	1120	4929	78.42%
Instance 11	29	41	10	1125	2275	77.73%

The generated instances were created, in part, to investigate the impact, if any, of instance characteristics on the difficulty of constructing high-quality solutions and the structure of high-quality solutions. The generated instances have different spatial characteristics, different ratios of number of suppliers to number of distribution centers, and different in- and outbound shipment patterns. More specifically, in **Instance 5** all facilities are in a small geographic region, in **Instance 6** all facilities are within a 200 mile radius of a CD, and in **Instance 7** all facilities are more than 200 miles away from a CD. The other generated instances have spatial characteristics similar to the three instances based on real-life data from a large U.S. retailer, but varying ratios of number of suppliers to number of distribution centers, and varying numbers of in- and outbound shipments at facilities (see Table 1).

The direct distance for a shipment, i.e., from the supplier at its origin to the distribution center at its destination ranges from 13 miles to 3,450 miles, with an average distance of 1,413 miles. The average per trailer per mile FTL lane rate, over the origin-destination pairs for which a LTL lane rate is specified, is \$13.21, and the average per trailer per mile FTL lane rate, over the origin-destination pairs corresponding to the shipments, is \$1.43.

A maximum out-of-route distance ratio of $M = 2.0$ is used and the charge for including an additional stop between i and j is set to $s^T(i, j) =$ \$50, for all lanes $(i, j) \in A^T$. When generating routes, we restrict the number of unique supplier (origin) locations and the number of unique distribution center (destination) locations visited by any single route; in this study, no more than five locations of each type can be visited by any route. The parameter settings for the maximum out-of-route distance, stop-off charge, and number of unique stops for each location type are the ones used by a large U.S. retailer. Furthermore, a neighbor radius of $\rho = 1000$ miles is used. Initial experimentation showed that a larger neighbor radius consistently tended to produce greater savings. Therefore, the neighbor radius is set to a reasonably large value that is acceptable in practice.

For all experiments, we use cost savings ($\Delta Cost$) as the primary performance measure, where cost savings are measured in percentages relative to a baseline cost computed when each order is moved by LTL transportation. The value of a 1% cost savings is different for each instance since the number of shipments can vary significantly between instances. For the instances evaluated in this study, 1% cost savings can range from \$1,000 to \$16,000.

Since we do not enumerate all feasible routes, we also compute a simple upper bound on the possible cost savings for an instance that is independent of the set of routes generated. The bound determines the minimum possible cost incurred for moving each shipment using LTL or FTL transportation. Utilizing the lane rates and shipment information, the minimum possible FTL transportation cost can be defined given that a shipment served by FTL transportation must travel at least the distance between the origin and destination. Consequently, a lower bound on the transportation cost given $p = (o_k, d_k)$ is:

$$LB = \sum_{k \in K} \min \left\{ r^L(p) * w_k * \ell(p), r^T(p_{min}) * \frac{w_k}{W} * \ell(p) \right\},$$

where $r^L(p) * w_k * \ell(p)$ is LTL shipment cost and $r^T(p_{min}) * \frac{w_k}{W} * \ell(p)$ represents the minimum possible cost utilizing FTL transportation for shipment k. For the FTL transportation cost, $\frac{w_k}{W}$ represents the fraction of vehicle capacity that shipment k requires, $\ell(p)$ is the distance between the origin and destination of shipment k, and $r^T(p_{min})$ is the minimum lane rate that shipment k can feasibly use. Shipment k can feasibly use a lane rate for (i, j) if the length of route $p = \{i, o_k, d_k, j\}$ is less than M times the distance between i and j. An upper bound on the maximum cost savings follows by subtracting this lower bound from the baseline LTL transportation cost. The maximum cost savings bound for each of the instances is reported in the last column of Table 1.

2.6.1 Route Generation

We first assess the importance of using transportation cost-based insertion as opposed to distance-based insertion (see Section 2.4). To do so, we conducted the following computational experiment. For each instance, we set $\tau = 1$ to limit each shipment to crossdock routes using only the crossdock that minimizes the out-of-route distance. We then generated two integer programs, one in which routes were generated using distance-based insertion and one where the routes were generated using transportation cost-based insertions. The final cost of each generated route for each IP is the true transportation cost computed via the approach in Section 2.3.2, independent of the insertion approach. Each IP was solved with a 4 hour time limit, and the resulting cost savings are presented in Table 2.

Table 2: Comparing Insertion Techniques

		$\Delta Cost$	
Instance	# Crossdocks (τ)	Trans. Cost Insertion	Distance Insertion
Instance 1	1	46.11%	45.34%
Instance 2	1	55.14%	50.60%
Instance 3	1	64.79%	60.95%
Instance 4	1	64.79%	61.19%
Instance 5	1	54.78%	52.81%
Instance 6	1	66.74%	63.23%
Instance 7	1	49.02%	45.75%
Instance 8	1	52.02%	47.61%
Instance 9	1	50.99%	46.42%
Instance 10	1	63.14%	59.40%
Instance 11	1	61.28%	58.20%

As expected, the transportation cost-based insertion procedure requires more computation time than distance-based insertions; the average time for route generation increased from 1.66 minutes to 5.48 minutes. However, the additional time spent on route generation is justified by the increased cost savings, as illustrated clearly by the results in Table 2. Therefore, in all other experiments the lane rate cost information will be used when determining insertion positions during route generation.

2.6.2 Shipment Plan Construction

The size of the integer program and the quality of a shipment plan are impacted by the number of crossdocks considered for each shipment. When this number increases, consolidation opportunities increase as well which may result in lower costs. In our first experiment, we test how solutions change as we increase the number of feasible crossdocks per shipment. Instance 1 has only a small number of shipments and the full IP with the entire route set R' can easily be solved, therefore it is an ideal candidate for this experiment. Since there are 14 CDs in Instance 1, the instance was solved 14 times, each time allowing one more crossdock to be considered for each shipment (in non-decreasing order of origin-destination distance via the crossdock). Each of the 14 resulting integer programs is solved to optimality in less than 30 minutes. Results summarizing the characteristics of the problems and optimal solutions are presented in Table 3. The table reports the number of crossdocks considered for a shipment (τ), the number of pickup-and-delivery routes R'_D generated, the number of crossdock inbound routes R'_I generated, the total number of crossdock outbound routes R'_O generated, the total number of routes generated, $\Delta Cost$, the percentage of shipments using LTL routes (LTL), FTL direct routes (FTL-D), and FTL routes visiting crossdocks (FTL-CD), the number of CDs used in the solution, and the time to solve the IP.

Table 3: Impact of the number of crossdocks considered for a shipment (Instance 1)

	# Routes Created					Shipment Breakdowns by %				Time
τ	R'_D	R'_I	R'_O	Total	$\Delta Cost$	LTL	FTL-D	FTL-CD	# CD Used	(Seconds)
1	808	193	696	1697	46.1%	71.1%	27.3%	1.6%	1	0.07
2	808	434	3132	4374	48.6%	73.4%	17.2%	9.4%	2	0.32
3	808	697	6795	8300	49.1%	73.4%	14.8%	11.7%	2	1.17
4	808	983	11394	13185	49.2%	68.0%	12.5%	19.5%	3	2.45
5	808	1300	18655	20763	50.5%	58.6%	5.5%	35.9%	3	5.71
6	808	1626	27180	29614	50.7%	57.8%	6.3%	35.9%	3	14.65
7	808	1972	36049	38829	51.2%	58.6%	5.5%	35.9%	2	35.73
8	808	2324	44983	48115	51.2%	58.6%	5.5%	35.9%	2	27.02
9	808	2693	56494	59995	51.7%	57.0%	6.3%	36.7%	4	29.87
10	808	3042	68754	72604	51.7%	57.0%	6.3%	36.7%	4	34.55
11	808	3387	81253	85448	51.7%	57.0%	6.3%	36.7%	4	114.68
12	808	3740	95479	100027	51.7%	57.0%	6.3%	36.7%	4	133.24
13	808	4065	110056	114929	51.7%	57.0%	6.3%	36.7%	4	483.30
14	808	4280	118876	123964	51.7%	57.0%	6.3%	36.7%	4	1475.74

As expected, the cost savings increase as the number of crossdocks considered for an order increases; from 46.1% when only a single crossdock is considered for an order to 51.7% when nine crossdocks are consider for an order (the maximum possible cost savings for this instance is bounded from above by 67.6%). The increased cost savings arise because the percentage of orders that is consolidated and shipped via a crossdock increases with the number of crossdocks considered for an order. Observe that this sometimes results in an increase in orders that are shipped using an LTL route (*e.g.*, when the number of crossdocks considered for a shipment is increased from 1 to 2). This happens when orders that were previously shipped together on a direct FTL route are no longer shipped together, because shipping some, but not all, via a crossdock creates better cost-saving consolidation opportunities. A similar phenomenon occurs when the number of crossdocks considered for a shipment increases from 6 to 7 and the number of crossdocks used in the optimal solution decreases. Because of the increase in the number of crossdocks considered for a shipment, one of the crossdocks is now a feasible option for many more orders and offers substantially better cost-saving consolidation opportunities. All of the shipments that were previously consolidated at another crossdock are now moved through this new crossdock, thus decreasing the total number of crossdocks used by one.

To see if relaxing the default parameter settings allows even larger cost savings, we performed an additional experiment using Instance 1 in which the number of crossdocks considered for each shipment is set to 14 and the neighbor radius is set to $\rho = 4000$ miles, which essentially removes all restrictions for this instance. The integer program solved in 69 minutes with a cost savings of 52.4%, a minor improvement. This result hints at the relative weakness of the maximum cost savings bound. It is also important to note that this result depends on the full set of routes R' produced by our route generation strategy, and that some additional cost savings may be found by considering additional feasible routes.

Instances 2 through 11 have a larger number of shipments and locations, and therefore many more routes are generated. Our second experiment focuses on finding high quality solutions to such larger instances. For these, it is rarely possible to solve the full IP using the entire set of routes R' in less than four hours (the imposed time limit). The heuristic search scheme using restricted route sets was developed for such larger instances, and its performance is analyzed in this experiment. Table 4 compares results obtained by solving these instances using the full IP and using the heuristic search scheme, where each approach is limited to four hours of compute time. The table contains the following columns: instance number, τ, the number of pickup-and-delivery routes R'_D generated, the number of crossdock inbound routes R'_I generated, the number of crossdock outbound routes R'_O generated, the total number of routes generated, and information pertaining to the best solution produced by the full IP and the best solution produced by the heuristic search scheme. For each of the solutions, there is a column for each of the following quantities: $\Delta Cost$, the percentage of orders shipped using LTL routes, FTL direct routes (FTL-D), and FTL routes visiting crossdocks (FTL-CD), the number of CDs used, and the solution time. The final optimality gap is also reported for the full IP approach. Recall that the maximum cost savings bound for these instances is reported in Table 1.

The results demonstrate the robustness of the heuristic search scheme. For all instances and for all considered values of τ, it finds a high-quality solution with either the same or larger cost-savings than the solution produced when solving the full IP (in the time limit of 4 hours). When the total number of routes in an instance is more than 150,000, solving the full IP can no longer reliably produce high-quality within the time limit (see *e.g.*, **Instance 3** with $\tau = 3$, **Instance 9** with $\tau = 2$ and $\tau = 3$, and **Instance 11** with $\tau = 2$ and $\tau = 3$).

Similar to what we have seen for **Instance 1**, when the number of crossdocks considered for an order is increased the cost savings and the percentage of orders shipped via a crossdock almost always increases. In contrast to Instance 1, however, we see that the increased density of shipments and locations in these instances leads to solutions when $\tau = 1$ with large fractions of shipments moving via truckload routes that do not involve a crossdock (column FTL-D). The existence of good consolidation routes without crossdocks in part mitigates the cost savings that results from increasing τ.

When τ is increased, the increase in the percentage of orders shipped via a crossdock is especially noticeable for Instance 4 and Instance 8, where where for $\tau = 3$ we see that 62.4% and 62.0% of orders are shipped via a crossdock, respectively. In Instance 4 there are

Table 4: Instance Results

						Full IP							Heuristic Search with Restricted IPs					
	#CD	# Routes Created						Shipment Breakdowns by %			#CD	Time		Shipment Breakdowns by %			#CD	Time
Inst.	(τ)	R'_D	R'_I	R'_O	Total	$\Delta Cost$	Opt Gap	LTL	FTL-D	FTL-CD	Used	(Hrs)	$\Delta Cost$	LTL	FTL-D	FTL-CD	Used	(Hrs)
2	1	6055	6778	3077	15910	55.1%	0.01%	52.7%	34.3%	13.0%	6	0.005	55.1%	52.7%	34.1%	13.2%	7	0.03
2	2	6055	24269	11103	41427	56.4%	0.96%	48.1%	28.2%	23.7%	8	4.00	56.5%	47.5%	27.6%	24.9%	8	4.00
2	3	6055	57087	25040	88182	57.0%	5.00%	33.1%	17.6%	49.3%	8	4.00	57.7%	44.4%	20.9%	34.7%	6	4.00
3	1	20778	13930	21736	56444	64.8%	2.60%	20.3%	49.5%	30.2%	9	4.00	65.3%	21.0%	46.4%	32.6%	10	4.00
3	2	20778	43111	99518	163407	62.2%	14.05%	15.5%	29.0%	55.5%	10	4.00	65.7%	21.8%	46.1%	32.1%	6	4.00
3	3	20778	88911	234269	343958	30.0%	55.42%	73.7%	26.3%	0.0%	0	4.00	66.2%	20.2%	38.1%	41.7%	7	4.00
4	1	6579	7092	2028	15699	64.8%	0.01%	25.2%	48.7%	26.0%	9	0.03	64.8%	25.2%	48.7%	26.0%	9	0.05
4	2	6579	22735	7170	36484	66.5%	0.56%	26.2%	34.0%	39.8%	9	4.00	66.5%	24.9%	35.4%	39.8%	10	4.00
4	3	6579	46903	15455	68937	66.3%	3.89%	17.9%	25.6%	56.5%	8	4.00	66.3%	16.3%	21.3%	62.4%	6	4.00
5	1	6205	3923	5263	15391	54.8%	0.00%	35.7%	56.6%	7.7%	4	0.00	54.8%	35.7%	56.6%	7.7%	4	0.01
5	2	6205	15474	24482	46161	55.2%	1.30%	29.3%	43.4%	27.2%	4	4.00	55.3%	31.1%	47.6%	21.3%	5	4.00
5	3	6205	34430	64540	105175	55.1%	5.40%	15.4%	33.7%	50.9%	4	4.00	56.2%	18.8%	42.2%	39.1%	4	4.00
6	1	13530	12767	7117	33414	66.7%	2.12%	20.8%	44.6%	34.6%	8	4.00	66.9%	22.2%	46.9%	30.9%	6	4.00
6	2	13530	41587	39577	94694	61.3%	21.09%	35.5%	63.6%	1.0%	3	4.00	68.0%	22.7%	35.9%	41.3%	5	4.00
6	3	13530	91077	102901	207508	67.3%	9.92%	12.4%	17.5%	70.1%	5	4.00	68.2%	23.9%	36.9%	39.2%	5	4.00
7	1	1159	916	1560	3635	49.0%	0.00%	48.8%	37.4%	13.8%	6	0.0001	49.0%	48.8%	37.4%	13.8%	6	0.001
7	2	1159	2705	5491	9355	51.1%	0.01%	47.6%	36.2%	16.1%	6	0.002	51.1%	47.6%	36.2%	16.1%	6	0.01
7	3	1159	5615	12258	19032	51.8%	0.01%	42.9%	24.4%	32.7%	7	1.29	51.8%	42.9%	24.4%	32.7%	7	4.00
8	1	7981	8420	8765	25166	52.0%	0.96%	44.2%	31.5%	24.2%	5	4.00	52.0%	42.1%	33.8%	24.1%	4	4.00
8	2	7981	28016	45954	81951	52.2%	8.14%	32.5%	17.2%	50.2%	5	4.00	54.2%	38.2%	24.1%	37.7%	4	4.00
8	3	7981	62187	114736	184904	52.4%	13.27%	25.9%	6.0%	68.1%	4	4.00	55.0%	28.3%	9.8%	62.0%	5	4.00
9	1	15235	17667	19597	52499	51.0%	2.49%	39.7%	26.8%	33.6%	9	4.00	51.4%	42.6%	26.8%	30.6%	9	4.00
9	2	15235	69478	109479	194192	43.2%	21.99%	54.0%	44.1%	1.9%	3	4.00	51.0%	46.9%	35.4%	17.7%	5	4.00
9	3	15235	158058	287729	461022	39.3%	31.95%	54.6%	44.0%	1.5%	3	4.00	52.3%	38.4%	20.0%	41.6%	5	4.00
10	1	21457	20905	15193	57555	63.1%	4.18%	23.1%	43.4%	33.5%	9	4.00	64.1%	23.7%	46.4%	29.9%	9	4.00
10	2	21457	72493	82750	176700	52.4%	29.98%	31.7%	45.0%	23.3%	8	4.00	64.4%	28.9%	48.4%	22.7%	5	4.00
10	3	21457	162163	206836	390456	50.6%	34.43%	33.0%	37.3%	29.6%	10	4.00	64.8%	28.7%	46.3%	25.1%	4	4.00
11	1	19699	13121	20322	53142	61.3%	6.07%	22.7%	41.7%	35.6%	9	4.00	62.9%	23.4%	41.9%	34.8%	7	4.00
11	2	19699	40557	103014	163270	49.3%	32.10%	32.4%	46.0%	21.6%	7	4.00	63.5%	25.2%	44.0%	30.8%	5	4.00
11	3	19699	83916	261174	364789	28.9%	52.95%	77.6%	22.4%	0.0%	0	4.00	63.5%	24.5%	41.7%	33.8%	6	4.00

relatively few distribution centers and each of them receives many shipments (each receiving about 7% of the shipments). As a result, there are many consolidation opportunities with $\tau = 3$, especially from a crossdock to a distribution center. The situation is almost completely opposite in Instance 8, because there are relatively few suppliers and all of them dispatch many shipments (all suppliers send about 2.5% of the shipments). As a result, there are many consolidation opportunities, especially from supplier to crossdock.

It is also interesting to note that in the solution for $\tau = 3$ for each of the instances only about half of the available crossdocks are used. This demonstrates the important impact on total freight transportation costs that results from choosing the right number and location of crossdock facilities.

2.6.3 Rate Differences

Since the goal of the optimization is to reduce freight transportation costs by exploiting FTL lane rates, it is clear that the difference between the LTL and FTL lane rate impacts the possible cost savings and the solution.

In our final experiment, we explore this impact by increasing the cost of LTL transportation for each shipment by 50%, thus making FTL transportation more attractive. When increasing the LTL transportation cost, the maximum savings possible for each instance also changes; the updated bounds on cost savings are found in Table 5. The results for the experiment are reported in Table 6. For each instance, we present the instance number, τ, columns pertaining to the solution for the original setting, columns pertaining to the solution for the new setting, and the decrease in the number of shipments dispatched on LTL routes (ΔLTL). All instances were solved using the heuristic search scheme that solves restricted IPs.

As expected, we see that when LTL transportation costs increase and FTL transportation becomes more competitive, the percentage of orders shipped on LTL routes is reduced and cost savings increase. A closer examination of shipment breakdowns reveals that in most cases the improvement comes from shifting orders from LTL routes to FTL routes

Table 5: Updated Bounds for Increased LTL Costs

Instance	Max Cost Savings Bound
Instance 1	78.26%
Instance 2	85.57%
Instance 3	85.76%
Instance 4	85.22%
Instance 5	87.85%
Instance 6	87.04%
Instance 7	81.92%
Instance 8	83.89%
Instance 9	83.95%
Instance 10	85.61%
Instance 11	85.15%

Table 6: Impact of Increased LTL Costs

		Original LTL Costs					Increased LTL Costs					
	#CD		Shipment Breakdowns by %			Time		Shipment Breakdowns by %			Time	
Inst.	(τ)	$\Delta Cost$	LTL	FTL-D	FTL-CD	(Hrs)	$\Delta Cost$	LTL	FTL-D	FTL-CD	(Hrs)	ΔLTL
1	3	49.1%	73.4%	14.8%	11.7%	0.0003	59.6%	40.6%	13.3%	46.1%	0.0004	32.8%
2	3	57.7%	44.4%	20.9%	34.7%	4.00	69.5%	29.2%	20.3%	50.5%	4.00	15.2%
3	3	66.2%	20.2%	38.1%	41.7%	4.00	76.1%	14.5%	42.1%	43.4%	4.00	5.7%
4	3	66.3%	16.3%	21.3%	62.4%	4.00	76.5%	11.3%	22.9%	65.8%	4.00	5.0%
5	3	56.2%	18.8%	42.2%	39.1%	4.00	69.7%	12.9%	47.0%	40.1%	4.00	5.9%
6	3	68.2%	23.9%	36.9%	39.2%	4.00	77.9%	9.5%	36.3%	54.2%	4.00	14.4%
7	3	51.8%	42.9%	24.4%	32.7%	4.00	65.3%	32.3%	28.7%	39.0%	4.00	10.6%
8	3	55.0%	28.3%	9.8%	62.0%	4.00	67.4%	18.0%	8.5%	73.5%	4.00	10.2%
9	3	52.3%	38.4%	20.0%	41.6%	4.00	64.4%	22.8%	22.9%	54.2%	4.00	15.6%
10	3	64.8%	28.7%	46.3%	25.1%	4.00	74.6%	20.4%	49.9%	29.6%	4.00	8.2%
11	3	63.5%	24.5%	41.7%	33.8%	4.00	73.7%	16.0%	47.5%	36.5%	4.00	8.5%

that visit a crossdock. In several instances, we see that the increased consolidation occurring at crossdocks further results in a shift from direct FTL routes to FTL routes that visit a crossdock.

CHAPTER III

IMPROVED INTEGER PROGRAMMING BASED NEIGHBORHOOD SEARCH FOR LTL LOAD PLAN DESIGN

3.1 Introduction

A *load plan* for a less-than-truckload (LTL) carrier is a fundamental component of the design of its service network. The load plan specifies the sequence of terminals through which each shipment will be transferred en route from origin to destination; since the plan also implies which shipments will be moved together, it is a freight consolidation plan. Many origin-destination terminal pairs served by a carrier will not attract enough freight to fill a daily trailer for dispatch, and therefore an excellent freight consolidation plan is critical to a carrier's success. National LTL carriers in the US typically spend millions of dollars weekly on linehaul transportation between terminals determined by the load plan.

This chapter develops and tests a new heuristic solution method for solving a load plan design optimization problem, building on prior work in Erera et al. [19]. In that paper, the authors propose a detailed time-space network representation of freight routing and consolidation decisions that accurately models the trailer flows that result from a given load plan given modern LTL operations. LTL carriers today offer much faster service to customers, and as a result freight spends less time at transfer terminals waiting for dispatch. Therefore, the detailed network model includes multiple dispatch times daily from break-bulk terminals. For the national carrier that motivated this research, the resulting network can have up to 5,000 nodes and 800,000 trailer dispatch arcs; adding decision variables that represent commodity-specific arc or path freight flows for thousands of origin-destination commodities yields impractically large integer programming (IP) instances. Erera et al. [19] therefore opose an *integer programming-based local search* heuristic solution approach for the problem, which proves to be effective. Two types of load plans can be generated:

(1) traditional load plans, where the set of freight transfer paths inbound to each destination terminal forms a directed in-tree; and (2) day-differentiated load plans, where the destination in-trees can vary by day-of-week.

One of the primary drawbacks of the original approach is that the neighborhood used does not allow changes to freight transfer paths inbound to multiple destinations within a single iteration. In this chapter, we propose an approach that uses two neighborhood types that remedy this shortcoming. The first neighborhood type attempts to attract additional freight volume to a specific lane between two terminals, and the second type attempts to move freight away from a lane. A key to the success of the approach is that we choose which origin-destination freight to add or remove from the lane by solving a modified version of the full integer programming model for the instance. To speed up the solution time for individual iterations (which in turn allows more heuristic iterations within a time budget), we also experiment with a new approach for handling empty trailer movements where we remove trailer balance constraints (and associated variables) from the neighborhood IPs and replace with lower bounds. Computational results using instances from a large US LTL carrier show that the approach can find load plans with costs 6 to 7% lower than those used in practice, and can find 2.5 to 5% additional cost savings when compared to the original approach using the same time budget.

The remainder of this chapter is organized as follows. Section 3.2 defines the load plan design problem. Section 3.3 presents a brief literature review for related research. In Section 3.4, the model used to represent freight routing is outlined and an IP for the load plan design problem is presented. Section 3.5 presents the proposed solution approach, including the details of the IP-based local search and the mechanism for integrating empty trailer movement decisions. Section 3.6 presents results from the computational study conducted using data from a national LTL carrier. Finally, Section 3.7 concludes the chapter with a brief discussion on lower bounds for the load plan design problem.

3.2 Load Plan Design

The load plan design problem is to determine how freight is transferred from origin terminals to destination terminals. A load plan specifies a single transfer path for each origin-destination terminal pair, where the path is comprised of direct trailer moves from terminal to terminal (known as *directs*). For each direct, freight is loaded into a trailer at its "from" terminal and unloaded from that trailer at its "to" terminal with no intermediate handling.

Load plans are designed to minimize total *linehaul costs* while meeting constraints on customer service. Linehaul costs are comprised of the transportation costs for moving loaded and empty trailers and the handling costs for transferring freight between trailers at a transfer terminal. To better understand the context of load plan design, we now provide a brief overview of LTL linehaul operations.

An LTL linehaul network is comprised of two types of terminals: *end-of-line* (EOL) and *breakbulk* (BB) terminals. Both of these terminal types are cross-docking facilities where freight is transferred from inbound trailers to outbound trailers. Each terminal serves customers in a local pickup and delivery region around the terminal; this local operation (sometimes known as the city operation) is separate from the linehaul operation, and will not be considered further in this chapter. EOL terminals, or *satellites*, serve as the interface between the linehaul network and the city operation in their regions, transferring freight from linehaul trailers to city trailers for delivery and transferring pickup freight from city trailers to linehaul trailers for linehaul dispatch. BB terminals also serve the same function, but additionally allow transfer of freight from inbound linehaul trailers to outbound linehaul trailers; BB terminals are linehaul *hubs*.

City operation trailers typically arrive back at terminals with pickup freight in the early evening (for example, by 7 pm), ready for transfer into the linehaul network. Freight to be delivered by the city operation to customers should arrive inbound from linehaul by early morning (for example, by 8 am) so that it can be transferred to a city trailer for a delivery route.

LTL carriers offer service standards, measured in business days, for delivery between pairs of locations. Single day (or overnight) service indicates that freight picked up today will be delivered tomorrow; this implies that the freight must move through the linehaul network from the origin terminal to the destination terminal overnight between 7 pm today and 8 am tomorrow. Service standards of 1, 2, 3, 4, and 5 days are common, although carriers today move a large fraction of their freight on lanes with standards of 3 or fewer days.

Traditional load plans are such that the set of transfer paths inbound to a single destination terminal form a directed in-tree; see Figure 2 for an example. The in-tree constraint

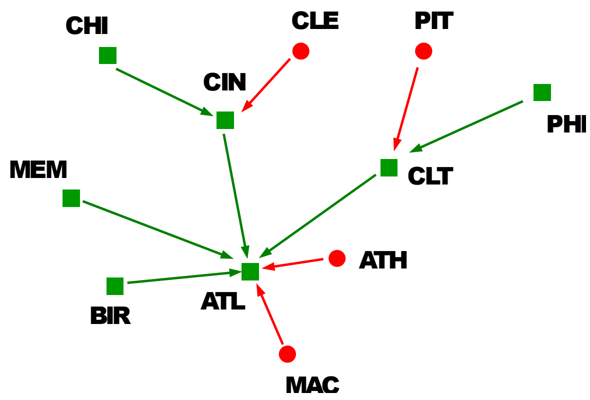


Figure 2: Example In-tree of Load Plan Paths to Atlanta; circles are satellite terminals, squares are hub terminals, and arrows are directs.

dictates that all freight at terminal i with destination terminal d must be transferred next to terminal j. An in-tree load plan is easy to operate in the field: dock workers at i move inbound freight destined for d, regardless of origin, into a trailer bound for terminal j. In addition to the in-tree constraint, LTL carriers may also wish to limit the number of transfer terminals that any shipment visits en route from its origin to destination.

Although an in-tree load plan is easy to implement, it is not usually trivial to determine the best such plan. Linehaul transportation costs are roughly proportional to trailer-miles (not ton-miles), and handling costs are proportional to the number of times freight is handled. Thus, excellent load plans seek to minimize trailer-miles without requiring excessive handling. A perfect world for an LTL carrier would be to attract enough customer freight at each terminal to load an integer number of full trailers outbound to every other terminal every day; in this case, the in-tree to each destination would be a star network. Such a system could offer the shortest possible service standards, while requiring the lowest possible transportation and handling cost. In practice, however, freight volumes vary from this ideal and linehaul freight from multiple origins (and to multiple destinations) must be consolidated into trailers for dispatch on directs. For example, in Figure 2, freight from Pittsburgh (PIT) and Philadelphia (PHI) bound for Atlanta (ATL) is sent first to Charlotte (CLT) where it is combined for dispatch to Atlanta. Not depicted, it is also likely that freight from these same two origins that is bound for destination Birmingham (BIR) is also sent first to Charlotte and then to Atlanta before final dispatch into Birmingham.

Finally, since freight volumes often have directional imbalances, an LTL carrier will also need to move empty trailers to reposition resources. Empty trailer movements increase trailer miles without carrying any freight. However, empty trailer movements also provide an opportunity in consolidation planning; in some cases, it can be cost-effective to plan to move certain origin-destination freight volumes along natural backhaul lanes where empty trailers would otherwise flow. It is therefore important to consider resource balance and empty trailer flows while determining a load plan.

3.3 Literature Review

The load plan design problem is a special case of service network design. For a discussion of related network design literature, see Erera et al. [19]. The service network design problem has been the focus of much research, and excellent reviews of the literature are provided by Crainic [14] and Wieberneit [52]. Service network design problems have been considered for a variety of freight transportation systems, such as trucking (*e.g.*, see [40], [42], [43], [41], [29], [12], [50], [6], [49], [19]), express shipment (*e.g.*, see [23], [8], [4]), freight rail (*e.g.*, see [38], [7], [55]), and multimodal transportation (*e.g.*, see [15], [39], [32], [28]).

Express shipment transportation service network design is to choose a set of cost-minimizing daily routes and schedules to transport shipments using a fleet of aircraft under strict time constraints. Barnhart and Schneur [9] presented a model and column generation approach for a single-hub express shipment service network design problem with specified time intervals for each shipment. Kim et al. [32] and Barnhart et al. [8] extended the work to multiple hubs. Both approaches decompose the problem into two subproblems: route generation and shipment movement. The key differences in express shipment service network design when compared to LTL network design are the fixed aircraft fleet with a varied composition of aircraft types, restricted capacity at hub locations, and a much smaller hub network. Instances evaluated in express shipment literature use at most nine hub locations, whereas the network for our load plan design problem has 58 facilities operating as hubs. Because of the network size difference and strict capacity limitations, approaches for the express shipment service network design are not appropriate for LTL network design.

In the rail industry, the primary service network design problem is often referred to as the blocking problem. The blocking problem is to choose which blocks (a group of railcars with a specified destination) to build at each yard, and to assign to each railcar a sequence of blocks to transfer it from its origin to its destination. Blocks are moved by the schedule of trains, and an effective block design minimizes total transportation, handling, and delay costs. Newton et al. [38] modeled the blocking problem as a network design problem where yards are represented by nodes and blocks are represented by arcs. The problem is solved via a column generation, branch-and-bound algorithm. Barnhart et al. [7] considered a similar formulation for the problem, but developed a solution approach using a dual-based Lagrangian relaxation that decomposes the problem into two disjoint subproblems. The key difference in the blocking problem when compared to LTL network design is the strict capacity limitations at each of the rail yards that limit the maximum number of blocks and maximum car volume that each yard can handle. The networks evaluated in the literature are very large like the one evaluated in our load plan design problem, but the number of commodities considered is significantly smaller. Given the strict capacity limitations at the rail yards and the size of the instances evaluated, it is not clear how effective the approaches developed for the blocking problem would be if adapted to accommodate instances seen in the LTL network design literature with a substantially larger number of commodities.

The load plan design problem for LTL carriers was introduced by Powell [40]; related work using the same flat network model is presented in Powell and Sheffi [42], Powell and

Sheffi [43], and Powell and Koskosidis [41]. Initial work used a flat (static) network to model the problem, and modeled the incentive for consolidation by prescribing that a minimum number of weekly trailers need to be dispatched on any direct lane used in the load plan. Powell [40] presented a local improvement heuristic that sequentially adds and drops directs to and from the plan. Our research uses a similar technique to drive the local search, but we use integer programs to find the best way to add freight or reduce freight on directs. More recent research has proposed metaheuristics for load plan design; Barcos et al. [6] presented an Ant Colony Optimization algorithm for the problem, and demonstrated the practical benefits of the approach on a small network with 49 EOL and 6 BB terminals.

Most recently, research has focused on using integer programming in load plan design. Jarrah et al. [29] presented a modeling scheme that decomposes a load planning integer program into separate IP subproblems for each terminal destination. A solution approach based on the decomposition and a slope-scaling technique produced high-quality solutions for large-scale problem instances from industry. The load plan design work by Erera et al. [19] is most closely related to our work in this chapter. The authors construct a detailed time-space network model of freight routing with a fine discretization of time to accurately represent consolidation opportunities; we adopt the same model in this research. The model also simultaneous routes loaded and empty trailers, modeling opportunities for the load plan to move freight along backhaul lanes. The paper also proposed a heuristic search that uses integer programs to find improving neighbors; using a neighborhood that allows changes to the load plan inbound to a single destination, the paper demonstrated that the technique can find substantial load plan improvements for large-scale networks with over 100 EOL and 50 BB terminals.

The IP-based neighborhood search heuristic developed in this chapter and used in Erera et al. [19] is a large-scale neighborhood search technique; see Ahuja et al. [1] for a survey. Solution approaches combining exact and heuristic search techniques have been proposed for many different problems, including the classic network design (*e.g.*, see [25]), vehicle routing (*e.g.*, see [18], [3], [22]), inventory routing (*e.g.*, see [45]), covering salesman (*e.g.*, see [44]), maximal covering (*e.g.*, see [26]), and multidimensional knapsack problem (*e.g.*, see [27]). One common IP-based neighborhood search framework solves a restricted version of the full IP formulation during each neighborhood iteration; this approach is used in Erera et al. [19], Hwang et al. [26], and Hwang et al. [27]. Our approach differs from that framework because we have designed a tailored IP to evaluate each neighborhood that is different than the traditional load plan design IP formulation.

3.4 Load Plan Design Problem and Model

The traditional load plan design problem considered in this chapter is to determine a set of triplets (i, d, j), where all freight arriving at terminal i (either from the city operation or inbound from the linehaul network) that is destined for terminal d is loaded next onto a direct trailer dispatched to terminal j. To do so, we use the model presented first in Erera et al. [19], which uses a time-space network representing trailer dispatch and freight transfer activity for a one-week planning horizon. Binary decision variables are used to select a unique terminal j for each pair i, d. Separate binary variables are used to select a unique time-space freight transfer path for freight moving between origin terminal o and destination terminal d each weekday, where each path is consistent with the load plan triplets.

For completeness, we now present the model in detail. Let $LN = (U, L)$ represent the carrier's linehaul terminal network, where U is the set of terminals in the network and L represents the set of potential directs connecting the terminals; triplet (i, d, j) can be part of a load plan only if $(i, j) \in L$. The linehaul network is mapped to a time-space linehaul network denoted $TS - LN = (N, A)$, where N represents the set of nodes and A the set of arcs. Each arc $a = (n_1, n_2) = ((u_1, t_1), (u_2, t_2)) \in A$ denotes a potential freight dispatch from terminal u_1 at time t_1 on direct $(u_1, u_2) \in L$ arriving at terminal u_2 at time t_2. If $u_1 = u_2 = u$, such an arc represents holding freight at u. Such holding arcs exist between all consecutive time-space nodes for u. The network $TS - LN$ is also "wrapped", such that the holding arc from the last time period is connected back to the first time period for each u, and that dispatch arcs near the end of the week wrap back to connect to nodes early in the week.

Freight demand, measured in fractional trailerloads, is assumed to be known for all pairs of (origin, destination) terminals for each weekday. Freight demand at an origin terminal $u_1 \in U$ on weekday d_1 is assumed to be available simultaneously from the city operation at a single time point in the evening of d_1. Freight that must reach destination terminal $u_2 \in U$ on weekday d_2 (based on service standards) must arrive at a single time point in the morning of d_2. Freight demand to be shipped from (u_1, t_1) to (u_2, t_2) is referred to as a *commodity*, where K is the set of all such commodities. For $k \in K$, let the origin terminal now be denoted $o(k)$ instead of u_1, and the destination terminal now be $d(k)$ instead of u_2.

For each $k \in K$, let $P(k)$ be the set of feasible paths in $TS - LN$ that connect $(o(k), t_1(k))$ to $(d(k), t_2(k))$; each such path is comprised of arcs in A representing dispatch or freight holding. A freight path $p \in P(k)$ is a sequence of time-space nodes: $p = \{(o(k), t_1(k)), (u', t_1(k)+t'), (u'', t_1(k)+t'+t''), ..., (d(k), t_2(k))\}$. A freight path is feasible only if it is consistent with the load plan; for the example path, the triplet $(o(k), d(k), u')$ would need to be included in the plan indicating that freight from $o(k)$ to $d(k)$ is loaded first to u'. Additionally, each freight path may be constrained to have a maximum number of intermediate terminals $\{u', u'', ...\}$. For each path p, the total handling cost h_p per pound is calculated by summing the costs of handling at the intermediate terminals visited.

We now provide some additional notation. Let $\Delta^+(u) \subseteq L$ denote the set of potential outbound directs from terminal $u \in U$, let $\ell(a)$ denote the direct $\ell \in L$ corresponding to the arc $a \in A$, let $\delta^+(n) \subseteq A$ denote the set of outbound arcs from node $n \in N$, let $\delta^-(n) \subseteq A$ denote the set of inbound arcs to node $n \in N$, and let c_a denote the per-trailer transportation (dispatch) cost along arc $a \in A$. For each commodity $k \in K$, w_k is the freight weight in pounds and q_k is the total freight size measured in fractional trailers (note, q_k need not be less than one).

For decision variables, let x indicate whether commodity k uses path p, *i.e.*, $x_p^k \in \{0,1\} \ \ \forall k \in K, \ \ p \in P(k)$. Let y represent the selected load plan by indicating whether direct $\ell \in \Delta^+(u)$ is used for outbound dispatch from u for all commodities destined for terminal d, *i.e.*, $y_\ell^d \in \{0,1\} \ \ \forall d \in U, \ \ \ell \in \Delta^+(u), u \in U$. Finally, let τ variables count the number of trailers (empty or loaded) that move on arc a, *i.e.*, $\tau_a \in \mathbb{Z}_+ \ \ \forall a \in A$.

The load plan design integer programming formulation ($TSLP-IP$) is then to minimize

$$\sum_{a \in A} c_a \tau_a + \sum_{k \in K} \sum_{p \in P(k)} h_p w_k x_p^k$$

subject to

$$\sum_{p \in P(k)} x_p^k = 1 \ \ \forall k \in K \tag{6}$$

$$\sum_{\ell \in \Delta^+(u)} y_\ell^d \leq 1 \ \ \forall u \in U, \ \ \forall d \in U \tag{7}$$

$$\sum_{p \in P(k): a \in p} x_p^k \leq y_{\ell(a)}^{d(k)} \ \ \forall k \in K, \forall a \in A \tag{8}$$

$$\sum_{k \in K} \sum_{p \in P(k): a \in p} q_k x_p^k \leq \tau_a \ \ \forall a \in A \tag{9}$$

$$\sum_{a \in \delta^+(n)} \tau_a - \sum_{a \in \delta^-(n)} \tau_a = 0 \ \ \forall n \in N \tag{10}$$

The objective is to minimize the total transportation and handling costs, where transportation costs are proportional to the integer number of trailers dispatched on each arc. Constraints (6) ensure that a path is chosen for each commodity. Constraints (7) ensure that a single outbound direct is selected for each terminal u and freight destined for terminal d. Constraints (8) ensure that a path can only be chosen for commodity k when all of its component directs are selected. Constraints (9) ensure that there are enough trailers moved along an arc to carry the freight assigned to the arc via the paths chosen. Finally, constraints (10) ensure flow balance of trailers at every node in the time-space network, and thus ensure proper repositioning of trailers.

3.4.1 Time-Space Network

We follow the same approach as in Erera et al. [19] to convert a linehaul network LN into a time-space version $TS-LN$; readers who desire more detail about the modeling choices are directed to that reference. In short, a time discretization approach is chosen to accurately approximate trailer dispatching. Short service standards today more frequently necessitates that trailers are dispatched before they are full, and therefore it is less appropriate to use aggregated dispatch arcs that represent, for example, all trailers dispatched today between terminal pairs.

To create $TS - LN$, for each $u \in U$ a number of timed copies (u, t) are included in N. For EOL terminals, a copy is generated each weekday morning at t =8:00 and evening at $t = 19 : 00$. For BB terminals, a copy is generated each weekday for all $t \in \{1{:}00, 3{:}00, 5{:}00, 8{:}00, 14{:}00, 19{:}00, 21{:}00, 23{:}00\}$. Freight (and trailer) holding arcs are generated in $TS - LN$ connecting consecutive timed node copies forward in time for each terminal u. A freight (and trailer) dispatch arc $((u_1, t_1), (u_2, t_2)) \in A$ is specified for each direct $\ell = (u_1, u_2) \in L$ outbound from u_1 for each of its timed copies $(u_1, t_1) \in N$. The arrival timed copy (u_2, t_2) is chosen to be the earliest next node such that $t_2 - t_1$ is not less than the travel time required by the trailer dispatch.

Freight entering the linehaul network at terminal u_1 on day d_1 is represented in $TS-LN$ as entering the network at $n_1 = (u_1, t_1)$, where $t_1 = d_1$ at 19:00. Freight that must reach its final destination at terminal t_2 on day d_2 is assigned the destination node in the time-space linehaul network of $n_2 = (u_2, t_2)$, where $t_2 = d_2$ at 8:00.

3.5 Solution Approach

Finding optimal solutions to instances of $TSLP - IP$ is difficult for the large-scale instances that occur in practice. The primary difficulty arises because the set of all feasible paths, $\cup_k P(k)$, is very large. Since the number of commodities $|K|$ is also usually large, there are also large numbers of constraints (8).

A possible solution approach would be to employ a branch-and-price algorithm and use column generation to dynamically generate new potential freight paths via a pricing subproblem. We opt for a simpler technique in this research that proves to be very effective: an integer programming-based local search methodology similar to that in Erera et al. [19]. In the IP-based local search approach, a restricted version of $TSLP - IP$ is solved during each local search iteration to identify (potentially) an improving neighbor. By fixing the paths for a large subset of the commodities K, these restricted IPs can be solved directly (optimally or near-optimally) using commercial IP solvers given a short compute time limit.

In Erera et al. [19], each restricted IP was generated by fixing the load plan variables y for all but a single destination d, and by also fixing all path selection variables x for each commodity k where $d(k) \neq d$. This approach has two primary limitations. First, the restricted IP for destination d is still a large NP-hard optimization problem which is often difficult to solve to a small optimality gap within a short computation time limit. This drawback creates unsatisfactory limits in practice on the number of local search iterations that are possible within a fixed time budget. Second, it is sometimes difficult to find improvements without jointly considering changes to the load planning in-trees for multiple destinations simultaneously; this is especially true for destinations in nearby proximity to one another, such as a breakbulk terminal and its nearby EOL satellites.

Our focus then is to create a search IP to employ within the IP-based local search that can simultaneously reroute freight to multiple destinations, but that has a reduced size that allows faster solve times per iteration.

3.5.1 NewOrOldTree IP-Based Local Search

The approach we devise is denoted the *NewOrOldTree IP-Based Local Search*, and is outlined in Algorithm 2.

Algorithm 2: *NewOrOldTree* IP-Based Local Search

```
Require: an initial load plan y and timed paths x
  Set iter = 0
  while compute time has not exceeded a prespecified time limit T do
    Choose whether to attract or reduce freight
    Randomly choose a direct ℓ ∈ L
    if ∃ at least one destination to consider then
      Solve NewOrOldTree IP
      if an improved solution is found then
        Update the current solution y and x
      end if
    end if
    Set iter = iter + 1
    if iter exceeds threshold then
      Solve Empty Balance MCNF generating new Empties_a
      Update the best known feasible solution
      Set iter = 0
    end if
  end while
```

In most applications, an existing load plan is to be improved, so an initial plan y is known. Determining an initial set of timed paths x, one for each commodity $k \in K$ and consistent with the load plan, is less straightforward in practice. The simplest option would be to determine and select the path for each k that arrives earliest (thus, with minimum waiting at intermediate transfer terminals). Such a solution ignores the cost benefits that may result from waiting at transfer terminals. Our computational study in Section 3.6 presents an alternative approach.

The *NewOrOldTree IP-Based Local Search* creates a neighborhood each iteration using a single direct ℓ selected at random with equal probability from all directs in L. One neighborhood attempts to attract freight flow to direct ℓ, while the other neighborhood attempts to reduce freight flow on ℓ. Given ℓ and whether we are to attract or reduce freight, simple rules are used to define $D' \subset U$, the set of destination terminals whose load plan in-tree is a candidate to be altered using ℓ. When the search is to attract freight to ℓ, each $d \in D'$ is such that $y_\ell^d = 0$ currently (freight is not routed to d on ℓ) and that the new potential in-tree to d created by setting $y_\ell^d = 1$ (while simultaneously also deselecting y_v^d, the current tree arc for d outbound from the tail node of ℓ) is such that it defines only feasible new freight paths to d. When the search is to reduce freight, each $d \in D'$ is such that $y_\ell^d = 1$ currently and that an alternative feasible tree arc w exists for d from the tail node of ℓ. For each $d \in D'$, then, we have identified a feasible new in-tree that either contains ℓ or removes ℓ. More details covering the determination of D' and the new trees is presented is Sections 3.5.4 and 3.5.5.

The *NewOrOldTree* IP is used to jointly select either the current (old) or the new in-tree for each destination $d \in D'$, to minimize the same objective function of $TSLP - IP$ (the sum of trailer transportation costs and freight handling costs at transfer terminals).

3.5.2 *NewOrOldTree* Integer Program

The *NewOrOldTree* IP is used to search an attract-freight or reduce-freight neighborhood defined by a given direct ℓ. It is not a simple restriction of the complete load plan design integer program $TSLP - IP$. First, for each destination $d \in D'$, a new binary decision variable z_d is used to make a selection between the current (old) in-tree or the new in-tree for d. Thus, the variables y are eliminated and implied instead by z. Second, trailer flow balance constraints are removed and replaced by flow lower bounds as an approximation.

In addition to choosing between the old or new in-tree for each d via z_d, the *NewOrOldTree* IP must also select a specific timed path $p \in P(k)$ in $TS - LN$ for each commodity $k \in K' \subset K$, where K' is the set of all commodities where $d(k) \in D'$ and the new in-tree for $d(k)$ implies that the path p_{old} currently used for k is no longer compatible with the load plan. Rather than considering all possible feasible paths, let $P'(k, Tree(d(k)))$ be a subset of the timed paths in $P(k)$ that follow the sequence of directs in L from $o(k)$ to $d(k)$ specified by $Tree(d(k))$. In this research, $P'(k, Tree(d(k)))$ contains at most three timed mappings of this sequence, each beginning at $(o(k), t_1(k)) \in N$ and terminating at $(d(k), t_2(k)) \in N$ and thus are service feasible. The first path utilizes so-called *purchased transportation* when possible; at some points in time, two terminals may be connected with an additional arc $a \in A$ that represents movement of trailers by railroad or third-party trucking services. This path dispatches the freight as early as possible on each direct, and selects purchased transportation if available at this earliest dispatch time. The second and third paths never select purchased transportation arcs. The second path dispatches the freight as early as possible at each terminal stop, while the third path holds freight to the last possible feasible dispatch time at the terminal immediately prior to the destination $d(k)$.

We are now ready to formally specify the *NewOrOldTree* IP. Let $P'(k, NewTree(d(k)))$ be the set of timed paths for commodity $k \in K'$ compatible with the new in-tree for $d(k) \in D'$, and $P'(k, OldTree(d(k)))$ be the set compatible with current (old) in-tree. Let $P'(k) = P'(k, NewTree(d(k))) \cup P'(k, OldTree(d(k)))$. Let h_p denote the sum of the handling costs per pound required at the intermediate terminals visited by p, and let c_a denote the per-trailer travel cost along arc $a \in A' \subseteq A$; note that A' only contains arcs used in some timed path in the P' sets. For each arc $a \in A'$, let f_a be the fixed fractional trailers moving on arc a given the fixed paths $p(k)$ in the current solution for all commodities $k \in K \setminus K'$. Furthermore, let $MinTrailers_a$ specify a minimum number of trailers that should move on arc a, given that trailer flow balance is to be maintained; see Section 3.5.3 for a discussion. Finally, let fixed cost F capture all fixed costs for trailers moving on arcs

$a \in A \setminus A'$ and the handling costs for all commodities $k \in K \setminus K'$.

There are three sets of decision variables. First, binary x_p^k variables indicate whether path p is used for commodity k. Second, binary z_d variables indicate whether the new load plan in-tree is used for destination d. Finally, integer τ variables count the number of trailers (empty or loaded) that move on arc a.

The *NewOrOldTree* IP is to then minimize

$$\min F + \sum_{a \in A'} c_a \tau_a + \sum_{k \in K'} \sum_{p \in P'(k)} h_p w_k x_p^k \tag{11}$$

subject to

$$\sum_{p \in P'(k)} x_p^k = 1 \;\; \forall k \in K' \tag{12}$$

$$x_p^k \leq 1 - z_{d(k)} \;\; \forall k \in K', \;\; p \in P'(k, OldTree(d(k))) \tag{13}$$

$$x_p^k \leq z_{d(k)} \;\; \forall k \in K', p \in P'(k, NewTree(d(k))) \tag{14}$$

$$\sum_{k \in K'} \sum_{p \in P'(k)} q_k x_p^k + f_a \leq \tau_a \;\; \forall a \in A' \tag{15}$$

$$x_p^k, \;\; z_d \text{ binary} \tag{16}$$

$$\tau_a \geq MinTrailers_a \text{ and integer} \tag{17}$$

The objective function minimizes total transportation and handling costs. The set of constraints (12) requires that at least one path is chosen for each commodity in K'. Constraints (13) and (14) are included so that a path can only be used if it coincides with the load plan in-tree choice, old or new, for each destination. Constraints (15) sum the freight on each impacted arc in the time-space linehaul network to determine how many trailers are needed.

We next discuss our approach for approximating trailer flow balance, and then how we generate attract and reduce freight neighborhoods (defined by K' and P').

3.5.3 Approximating Trailer Balance

To keep the size of each *NewOrOldTree* IP relatively small, we do not include the trailer flow balance constraints (10). Instead, we rely on trailer flow lower bounds determined by periodic solution of an empty trailer repositioning minimum cost network flow (MCNF) formulation. Given a fixed load planning solution for *TSLP-IP* specified by x and y, constraints (9) can be used to determine the number of loaded trailers, τ_a^L, required for dispatch on each time-space arc a. Since the loaded flow on $TS-LN$ is not likely to be balanced, we can solve a MCNF for minimum trailer cost repositioning using c_a and node net supplies defined by τ_a^L. Let τ_a^E be the number of empty trailers moving on each arc $a \in A$ in this solution.

To provide an incentive for the *NewOrOldTree* local search to route freight in a way that leads to rough trailer balance, we add lower bounds on trailer flows based on the most recent empty repositioning solution. For each arc $a \in A'$ where there exists any empty trailer flow ($\tau_a^E > 0$), let $MinTrailers_a$ be defined as $\tau_a^L + \tau_a^E$, the total number of trailers moving on this dispatch arc at the time of the most recent empty repositioning solution. Since (17) forces τ_a to be greater than this bound, a fixed cost is created which guides the load plan to route freight along these backhaul dispatch lanes. We experimented with using a smaller bound of τ_a^E, but found better performance when using the sum of loaded and empty trailers. A partial explanation of this better performance can be provided by recognizing that loaded trailers moving on backhaul lanes are often very lightly loaded (*e.g.*, only 10-15 % full); the $MinTrailers_a$ bound as defined recognizes that many of the loaded trailers moving on a are already the result of freight being routed to backhaul lanes.

It is important to update the bounds $MinTrailers_a$ by solving the MCNF periodically during the local search, recognizing that empty trailer movements will change somewhat as the load plan changes. In Section 3.6, we experiment with the update frequency. In results that we do not report in this chapter, we experimented with an alternative *NewOrOldTree* IP that includes the full set of trailer balance constraints (10) instead of the lower bounds. We found that the total solution time required to complete 1000 iterations of the local search increased from 2 hours to more than 12 hours, with little improvement in the final load plan solution. Thus, the bounding approach seems pragmatic and effective.

3.5.4 Attracting Freight

We now discuss how to form the neighborhood defined by K' and P' during each iteration of the local search. First, we consider the attract freight version.

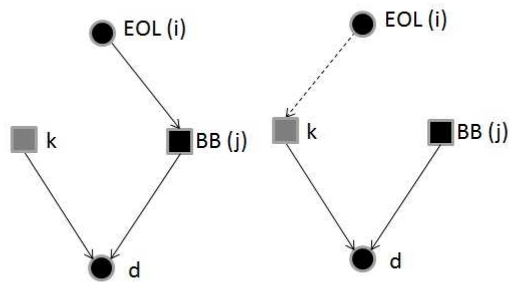
To attract freight to direct $\ell = (i, j) \in L$, we consider adding ℓ to the in-tree for each destination $d \in U$ for which it is not already included ($y_d^\ell = 0$). Replacing the current outbound direct (i, j') to d with (i, j) will affect the freight routing for all commodities $k \in K$ with $d(k) = d$ and whose path from $o(k)$ to d includes terminal i in the current in-tree. Note that this approach to determining potentially impacted commodities is consistent with the approach for the IFOL-0 procedure detailed in Powell [40].

To form K', however, we only allow replacing (i, j') with (i, j) in the in-tree for d if all affected commodities can be routed along the new freight path without violating service requirements. For each destination d where $y_d^\ell = 0$, we test each commodity k where $d(k) = d$ whose path to d currently includes i. The new path follows the existing in-tree from $o(k)$ to i, then transfers to j and follows the existing in-tree to d. If this path has total minimum travel and transfer time that is less than the requirement for commodity k and does not violate the constraint on maximum number of intermediate transfer terminals, then it is marked as feasible. Only if the new path for each such commodity k with destination d is feasible, then all such commodities are added to K' and d is added to D'. The new in-tree for d includes (i, j) replacing (i, j'), and the path set $P'(k, NewTree(d(k))$ is formed with the timed paths for each k following the new in-tree. We then repeat this process for each potential d.

3.5.5 Reducing Freight

Reducing freight from $\ell = (i, j)$ is a bit more complicated. Of course, it is easy to identify the potential commodities k that may be affected as simply those with destination $d(k) = d$ where (i, j) is included in the in-tree to d ($y_d^\ell = 1$) and the path from $o(k)$ to d includes direct (i, j). However, determining an appropriate new in-tree for destination d that excludes (i, j) requires selecting from potentially many feasible choices; it will not be possible in general to select the best choice. We outline our approach for doing so for three cases: (1) i is an EOL, and j is a BB terminal; (2) i is a BB, and j is an EOL; and (3) i and j are both BB terminals.

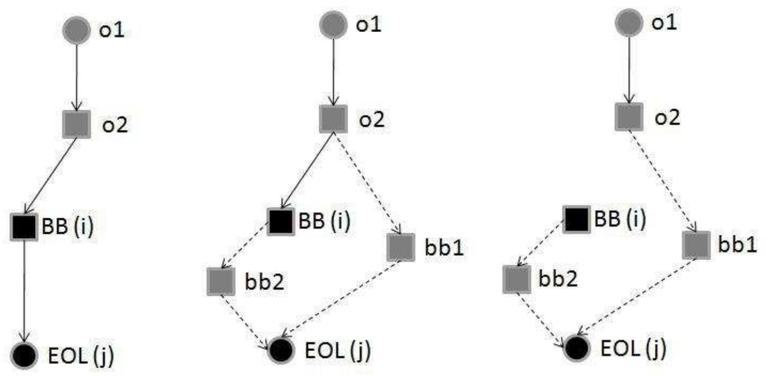
Case 1 is the simplest. Note that since i is an EOL, each potentially impacted commodity k has $o(k) = i$ and a different destination $d(k)$. Therefore, selecting a new in-tree for destination d that does not include (i, j) only impacts the single commodity k where $o(k) = i$ and $d(k) = d$. Figure 3 shows a simple example for this scenario. Figure 3(a) depicts the current load plan destination in-tree before freight is reduced on direct (i, j) and Figure 3(b) depicts a possible in-tree change.



(a) Current (b) Possible Change

Figure 3: Reducing Freight on an EOL-BB Direct

For Case 2 and Case 3, a new in-tree for d that does not include (i, j) must be feasible for potentially many commodities k, each with $d(k) = d$. Unlike the attract freight scenario and case (1) above, in these cases we allow multiple changes to the new in-tree to d in order to specify feasible paths for each commodity k. Figure 4 provides an example for case (2), where $d = j$. Figure 4(a) shows the current in-tree to j before attempting to reduce freight on (i, j). Figure 4(b) gives an example of an incompatible in-tree change that might be suggested if new paths for $(o1, j)$ and $(o2, j)$ are determined individually. Suppose $(o1, j)$ is proposed to follow path $(o1, o2, bb1, j)$ because the path $(o1, o2, i, bb2, j)$ violates feasibility (*e.g.*, suppose the maximum number of intermediate terminals is 2). Then, path $(o2, i, bb2, j)$ cannot be used for commodity $(o2, j)$ since the result would violate the in-tree constraint. Figure 4(c) provides an example of a compatible in-tree change. Note that several changes are made to the tree to create paths that avoid (i, j).



(a) Current (b) Incompatible Change (c) Compatible Change

Figure 4: Reducing Freight on a BB-EOL Direct

We now outline the specific approaches we use for each of three cases.

3.5.5.1 Case 1: EOL to BB

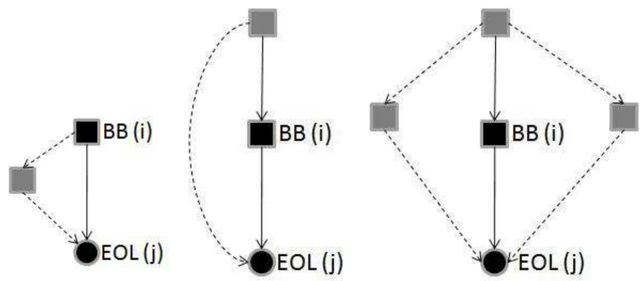
In Case 1, we consider one-by-one each destination d for which a commodity k exists where $o(k) = i$, $d(k) = d$, and (i, j) is in the in-tree to d. To do so, we identify an alternate terminal $j' \in U$ such that the path from i to j', and then following the existing in-tree to d has the lowest per trailer transportation and handling cost. If this path is feasible from the perspective of total time and maximum number of terminal transfers, then d is added to D', k is added to K', and $P'(k, NewTree(d(k)))$ is created with the timed paths for the new in-tree to $d(k)$ where (i, j') replaces (i, j).

3.5.5.2 Case 2: BB to EOL

In Case 2, since j is an EOL terminal then the only destination tree that is potentially impacted is the one with $d = j$. Let K'' be the set of commodities with $d(k) = j$ whose path over the current in-tree to j uses direct (i, j). We now define an approach for sequentially determining new paths from each $o(k)$ for $k \in K''$, such that all paths are feasible and form a new directed in-tree to j that does not include (i, j). We design the approach such that the current paths for k with $d(k) = j$ that are not in K'' remain feasible and compatible with the new in-tree.

We consider each $k \in K''$ in non-increasing order of q_k. Note that the set K'' defines all commodities with origins in a branch of the in-tree upstream from and including i. Given k, we create a number of path choices that are both feasible and compatible with choices made earlier for commodities in K'' that were already considered.

The first path choice follows the current in-tree from $o(k)$ to i, and then selects $j' \neq j$ and follows the current in-tree path from j' to j, where the selection of j' is determined by minimizing the per trailer transportation and handling cost of the resulting path from i to j. Figure 5(a) provides a sketch of this type of path. The second path choice (valid only for $o(k) \neq i$) follows the current in-tree to the predecessor of i for k, $\pi_i(k)$, and then skips i adding direct $(\pi_i(k), j)$ to the in-tree. This choice is usually feasible for k if the direct exists, since it reduces handling time and the number of intermediate transfers. Figure 5(b) provides a sketch of this type of path. The third path choice is generated by routing freight from $\pi_i(k)$ to an alternative BB terminal i' before onward transfer to j (again, valid only for $o(k) \neq i$). Such a path follows the current in-tree from $o(k)$ to $\pi_i(k)$, then adds the new direct $(\pi_i(k), i')$, then follows the current in-tree path from i' to j. Figure 5(c) provides a sketch of this type of path.



(a) Type 1 (b) Type 2 (c) Type 3

Figure 5: Reducing Freight on BB-EOL Directs: Potential Path Types

To select a path for k, the path choices are evaluated in order of non-decreasing cost until a service feasible (time and number of transfers) path is identified. If no path is identified, then the process terminates and this neighborhood is empty for ℓ. Once a path is selected for k, the load plan change required is fixed and cannot be altered when considering the remain commodities in K''. Note that this approach may result in new outbound directs selected only for terminals i and its predecessors in the new in-tree to d. If feasible new paths are found for all $k \in K''$, then $K' = K''$ and $P'(k, NewTree(d))$ is created with the timed paths using the new in-tree to d.

3.5.5.3 Case 3: BB to BB

Case 3 is similar to Case 2, but since j is a BB terminal, the in-trees to many destinations d in addition to j may be altered. Let D'' be the set of destinations where (i, j) currently is used on load plan in-tree to d ($y^{\ell}_{d} = 1$). For each $d \in D''$, we attempt to create a feasible new in-tree using a procedure similar but somewhat simpler than that for Case 2.

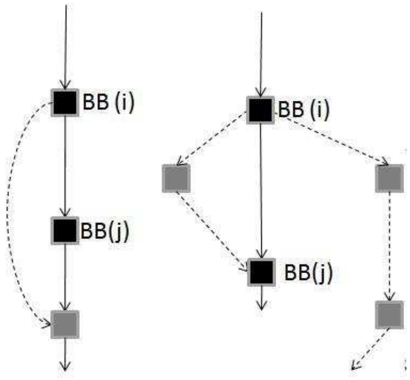
For a given $d \in D''$, let K'' again contain all commodities k with $d(k) = d$ where the path from $o(k)$ to d using the current load plan uses (i, j). Again, we will evaluate each $k \in K''$ in order of non-increasing freight volume q_k. In this case, however, we will only alter the outbound direct from node i, selecting a new next terminal $j' \neq j$. Once this choice is fixed for the commodity with largest freight volume, we check to see if this feasible for all remaining k in K''. If so, d is added to D', K'' is added to K', and $P'(k, NewTree(d))$ is created for each $k \in K''$ with timed paths using the new in-tree to d. Then, we work on the next $d \in D''$.

The first path choice for the largest volume commodity k follows the current in-tree from $o(k)$ to i, and then skips i by replacing (i, j) with $(i, \sigma(j))$ where $(j, \sigma(j))$ is the direct outbound from j in the current load plan. The choice does not exist when $d = j$. Figure 6(a) provides a sketch of this type of path. The second path choice is generated by routing freight from i to an alternative BB terminal j' before onward transfer to d. Such a path follows the current in-tree from $o(k)$ to i, then adds the new direct (i, j'), then follows the current in-tree path from j' to d. Figure 6(b) provides a sketch of this type of path.

To select the path for k, the generated path choices are sorted in non-decreasing order of total per trailer transportation and handling cost, and the lowest cost service feasible path is selected.

3.6 Computational Results

Algorithm 2 was implemented in C++ using ILOG Concert Technology and CPLEX 11 as the IP solver. When solving the IP formulations with CPLEX, all parameters were



(a) Type 1 (b) Type 2

Figure 6: Reducing Freight on BB-BB Directs: Potential Path Types

set to their default values. The experiments were run on a cluster of workstations with clockspeeds between 2.0 and 3.5 MHz all with 1.5Gb of memory.

Because a week-long planning horizon is typical in load planning, our test set contains six single week instances based on historical freight volume and terminal network data provided by a super-regional LTL carrier in the U.S. The instances represent six weeks of operations during 2010 and 2011 (Nov10-W4, Dec10-W1, Dec10-W2, Feb11-W3, Mar11-W4, May11-W1). Since some freight originating in one week may be due in the subsequent week, a "wrapped" version of the time-space network is used for planning. In this network, arcs connect later time periods in the week to time periods at the beginning of the week. The implicit assumption in a wrapped network is that identical freight volumes will repeat each week. Table 7 provides detailed characteristics for each of the instances used in this computational study. For these instances, a 1% linehaul cost savings represents about a $50,000 per week for the LTL carrier.

Table 7: Instance Characteristics

	# Terminals $\|U\|$		# Nodes	# Directs	# Arcs	# Commodities
Instance	# EOL	# BBs	$\|N\|$	$\|L\|$	$\|A\|$	$\|K\|$
Nov10-W4	102	58	5,489	24,181	842,287	60,231
Dec10-W1	102	58	5,489	24,181	842,287	60,307
Dec10-W2	102	58	5,489	24,181	842,287	59,736
Feb11-W3	102	58	5,489	24,181	842,287	59,043
Mar11-W4	103	58	5,503	24,492	849,859	62.259
May11-W1	103	58	5,503	24,492	849,859	60,852

We use our technology to improve a load plan provided to us by the LTL carrier.

Standard load plan information only specifies the in-tree structure, represented by y in our model. Therefore, it is necessary to determine dispatch paths for the freight commodities on the time-space network to complete an initial solution. In this research, the initial solution was generated using a tool developed by Zhang [54]. This approach selects a timed path consistent with the load plan for each freight commodity considered in the instance while taking into consideration driver operating constraints and empty trailer repositioning, with the objective of minimizing total linehaul cost. While there are simpler approaches for simulating the trailer dispatching induced by a load plan, they tend to overestimate costs and this lead us to overstate the gains.

In this computational study, we test various configurations of Algorithm 2 using the sample instances. All tests choose the direct ℓ in each iteration randomly with equal probability from the candidate set L; this equal probability approach performed better than biased alternatives in tests not included in this chapter. Experiments test the effectiveness of different strategies for selecting whether a given iteration should search an attract freight neighborhood or a reduce freight neighborhood. Specifically, we test the following variants:

1. Reduce freight only (no attract neighborhoods) (RO)
2. Attract freight only (no reduce neighborhoods) (AO)
3. Randomly alternate between reduce and attract freight, choosing attract freight with a 25% probability (Rand_0.25)
4. Randomly alternate between reduce and attract freight, choosing attract freight with a 50% probability (Rand_0.50)
5. Randomly alternate between reduce and attract freight, choosing attract freight with a 75% probability (Rand_0.75)

3.6.1 Solving Neighborhood IPs

For the IP-based neighborhood search to be successful, the search IP must yield a nearly-optimal solution in a short amount of computation time. To verify that the *NewOrOldTree*

IP is effective, we evaluated the *NewOrOldTree* IP performance within six hour runs of Algorithm 2 using the AO and RO variants. Both runs were conducted using the Nov10-W4 test instance. During the AO run, 1,897 instances of the *NewOrOldTree* IP were solved, and during the RO run, 3,910 instances were solved.

Table 8 presents summary statistics for the *NewOrOldTree* IP instances that were solved. We first note that the average number of destinations considered simultaneously for each IP is typically substantially larger than 1; recall that the neighborhood used in Erera et al. [19] includes all commodities k where $d(k) = d$, a single destination. It is also clear from the table that the search IP instances are relatively small by modern standards, on average requiring fewer than 10,000 integer variables and constraints.

Table 8: Neighborhood IP Instances

	# Destinations			# Constraints			# Variables		
IP Type	min	max	average	min	max	average	min	max	average
Attracting	1	140	64.7	97	47,969	6,914.3	64	38,076	5,076.8
Reducing	1	138	11.7	21	16,975	1,674.4	14	13,261	1,230.0

In this experiment, each search IP is given a computation time limit of 60 seconds. For freight attract instances, approximately 90 % of the IPs solve to a 0.01% optimality gap within the time limit. The remaining 10 % yield feasible solutions with a maximum optimality gap of 3.0 %. Furthermore, 56 % of the instances solve to optimality in less than 1 second, while 86 % solve in fewer than 30 seconds. For freight reduce instances, approximately 98 % of the IPs solve to optimality using the 0.01% gap within the time limit. Again, the remaining 2 % of the instances find a feasible solution with a maximum optimality gap of 3.0 %. In this case, 90 % of the instances solve to optimality in less than 1 second, while 97 % require fewer than 30 seconds. Clearly, the search IP is able to find excellent solutions with short solve times on practically-sized instances.

3.6.2 Empty Integration

In this section, we report on the effectiveness of the approach for approximating trailer balance constraints within the *NewOrOldTree* search IPs, and provide suggestions for a reasonable frequency at which to update the lower bounds $MinTrailers_a$; refer back to

Section 3.5.3 for a description of the approximation approach.

Recall that each update of $MinTrailers_a$ requires solving a MCNF empty repositioning optimization problem on the network $TS - LN$. Suppose we measure the update frequency using the number of local search iterations since the most recent update. Table 9 summarizes average final solution quality as a function of empty rebalancing frequency. Each table entry represents the performance metric averaged over 6-hour runs with a 60 second limit per search IP for the May11-W1 instance using each of the five solution strategies (AO, RO, Rand_0.25, Rand_0.5, Rand_0.75). Row $\Delta Cost$ reports the percentage cost savings of the final solution over the initial load plan, ΔPPT reports the percentage increase in pounds per trailer in the final solution versus the initial, and $Time$ reports the fraction of the total compute time devoted to solving empty repositioning MCNF models. The final column, labeled Single, provides as reference these same values for a simple strategy where a single empty balance MCNF is solved for the initial load plan and then not updated again until the end of the 6-hour run, when a final MCNF is solved to balance the final load plan.

Table 9: Average Solution Quality for Various Empty Rebalancing Update Frequencies

	Empty Rebalancing Frequency						
	1	10	20	50	100	200	Single (∞)
$\Delta Cost$	5.05	6.24	6.33	6.52	6.58	6.36	6.35
ΔPPT	8.58	11.58	11.54	12.23	12.07	11.93	11.75
$Time$	58.16	17.57	10.00	4.26	2.51	1.20	0.18

Table 9 clearly depicts a tradeoff between rebalancing frequency and solution quality. When rebalancing occurs too frequently, too much of the total computation time is spent on solving MCNF formulations and therefore the local search cannot make enough progress. When rebalancing occurs too infrequently, outdated empty estimates sometimes guide the search toward solutions with somewhat lower quality. We note, however, that since a large fraction of the empty trailer movement requirements are based only on freight demand and not on the load plan used to satisfy demand, the Single strategy does not perform that poorly on average. Figure 7 demonstrates that this general pattern holds also for specific solution strategies, by comparing the percentage cost savings generated by the RO, AO, and Rand_0.5 approaches. In this experiment, the best update frequency is every 100 local search iterations independent of approach, although the RO approach appears to generate similar results even when the trailer flow lower bounds are not updated at all during the local search.

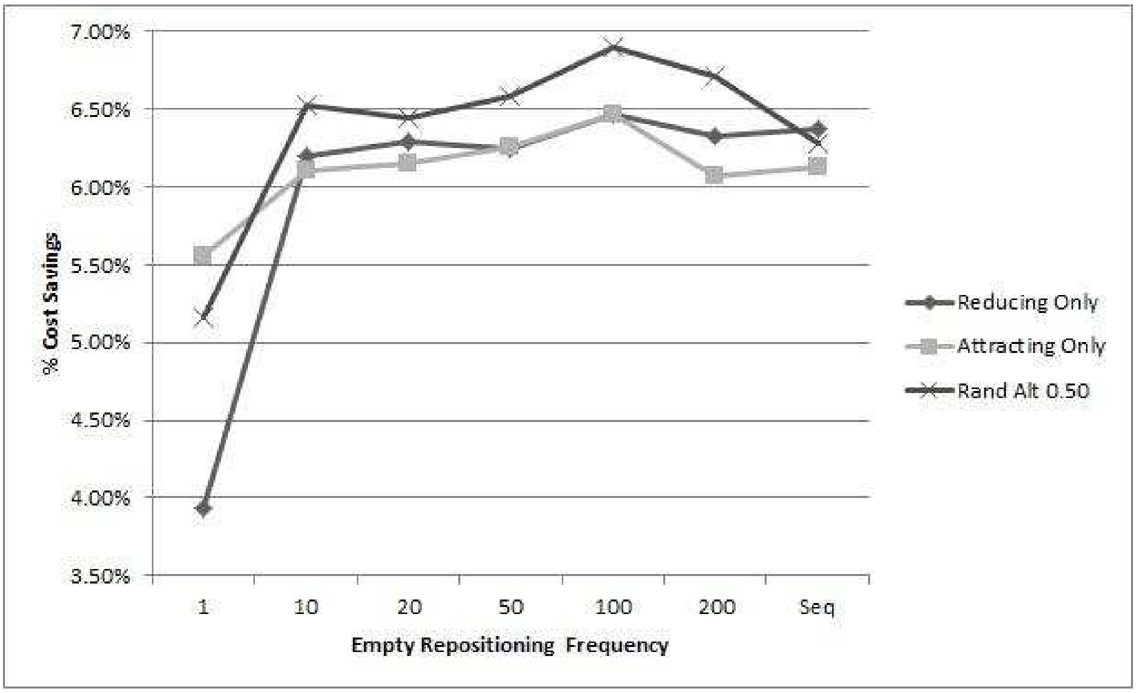


Figure 7: Percentage Cost Savings vs. Empty Repositioning Frequency

Based on these results, we will use an empty repositioning update frequency of 100 local search iterations for the remainder of experiments reported in this chapter. Additional experimentation was conducted to ensure that similar results were found when running each of the techniques for the local search heuristic on different instance weeks. We found consistent results, but we do not report details here.

3.6.3 Comparison of Neighborhood Search Variants

Experiments were conducted to compare five variants of the neighborhood search approach on multiple instances. Tables 10 and 11 present the percentage cost improvements from initial to best solution for each of the variants. Table 12 summarizes the performance of each variant by averaging the cost improvements across all instances. In each of these tables, the percentage cost savings ($\Delta Cost$) and percentage increase in pounds per trailer (ΔPPT) are reported. Since a 1% cost savings represents about \$50,000 per week, all techniques generate substantial improvements from the initial solution provided by the carrier.

Table 10: Performance of Neighborhood Search Variants for One Hour Runs

		RO	AO	Rand_0.25	Rand_0.50	Rand_0.75
Nov10-W4	$\Delta Cost$	3.15	3.88	3.35	3.92	4.10
	ΔPPT	4.62	5.77	5.43	5.65	5.68
Dec10-W1	$\Delta Cost$	2.85	3.98	3.65	3.43	4.08
	ΔPPT	4.48	6.00	5.19	5.75	6.19
Dec10-W2	$\Delta Cost$	2.84	4.08	3.51	3.77	3.89
	ΔPPT	4.17	5.99	5.19	6.26	5.82
Feb11-W3	$\Delta Cost$	3.22	4.54	4.02	4.38	4.00
	ΔPPT	4.81	6.42	5.75	6.28	5.55
Mar11-W4	$\Delta Cost$	3.30	3.76	3.63	3.91	4.01
	ΔPPT	4.30	5.06	5.13	5.29	5.72
May11-W1	$\Delta Cost$	2.73	4.03	4.08	3.96	3.94
	ΔPPT	4.02	5.92	6.08	5.26	5.41

Table 11: Performance of Neighborhood Search Variants for Six Hour Runs

		RO	AO	Rand_0.25	Rand_0.50	Rand_0.75
Nov10-W4	$\Delta Cost$	6.75	6.46	7.16	6.82	6.87
	ΔPPT	11.40	12.22	13.23	13.51	13.08
Dec10-W1	$\Delta Cost$	6.47	6.29	6.90	6.59	6.72
	ΔPPT	11.35	11.66	13.22	13.22	12.08
Dec10-W2	$\Delta Cost$	6.74	6.33	7.08	6.73	6.47
	ΔPPT	11.70	12.76	13.21	13.43	12.65
Feb11-W3	$\Delta Cost$	6.84	6.88	7.20	6.93	7.09
	ΔPPT	11.74	13.23	13.61	13.59	13.02
Mar11-W4	$\Delta Cost$	6.52	6.24	6.11	6.68	6.34
	ΔPPT	11.37	11.64	10.65	12.74	11.45
May11-W1	$\Delta Cost$	6.50	6.31	6.78	6.54	6.45
	ΔPPT	11.59	11.56	12.34	12.22	12.09

Table 12: Average Percent Savings Over All Instances

		RO	AO	Rand_0.25	Rand_0.50	Rand_0.75
1 hour	$\Delta Cost$	3.02	4.05	3.71	3.90	4.00
	ΔPPT	4.40	5.86	5.46	5.75	5.73
6 hours	$\Delta Cost$	6.64	6.42	6.87	6.72	6.66
	ΔPPT	11.53	12.18	12.71	13.12	12.40

Although all of the variants generate improvements, some perform better than others. When the variants are restricted to a one hour time limit, the attract only (AO) method performs best on average. In this setting, the reduce only (RO) technique appears to be dominated by approaches that include some freight attraction iterations. On the contrary, for the six hour runs, the reduce only (RO) variant performs slightly better than the attract only (AO). However, the approaches that use a mix of attracting and reducing freight iterations perform best, with the Rand_0.25 technique achieving the highest average cost savings.

When considering all of the computational results, the techniques that use some combination of attract freight and reduce freight neighborhoods provide the most reliable performance. The technique with the highest average cost savings summed for both the one hour and six hour runs is the Rand_0.75 variant. However, the Rand_0.25 and Rand_0.50 are quite competitive in this metric. In most pure load planning applications, a six hour compute time limit is reasonable, and therefore it seems reasonable to choose a mixed attract and reduce freight search strategy biased somewhat toward the reduce freight neighborhood.

3.6.4 Analysis of Load Plan Changes

To understand in part the source of the cost savings identified by our load plan improvement method, we analyze the differences in the freight and trailer flows between the best load plan found by our approach and the initial load plan for one of the test instance weeks. We focus our analysis on the final load plan produced for Feb11-W3 using the Rand_0.25 variant, since this technique and instance yielded the largest relative cost savings among all instances and technique variants.

Table 13: Change in Number of Transfers Used by Freight Flow

	0	1	2	≥ 3
Initial Load Plan	50.27%	40.35%	8.86%	0.52%
Improved Load Plan	48.22%	41.98%	9.24%	0.56%

Most of the improvement found by our local search is a result of increased freight consolidation, as evidenced by the 13% increase in pounds per trailer dispatch. Table 13 summarizes the percent of freight volume (fractional trailerloads of commodity demand) that transfers at 0, 1, 2, and $\geq$3 intermediate terminals in the initial and improved load plans respectively. The fraction of freight transferring 0 times decreases in the improved load plan, while the percent of freight transferring 1, 2, and $\geq$3 times increases, providing additional clear evidence of additional consolidation.

Table 14 shows how the total freight flow (measured in fractional trailers) is distributed across directs of different types in the initial and improved load plans, as well as the increase in pounds per trailer (ΔPPT) by direct type. The majority of freight flow volume moves

Table 14: Freight Flow Changes by Direct Type

	Percent of Total Freight Flow		
Direct Type	Initial Freight Flow	Improved Freight Flow	ΔPPT
BB-BB	61.9%	62.4%	12.05%
BB-EOL	22.8%	22.5%	15.73%
EOL-BB	15.1%	14.9%	8.83%
EOL-EOL	0.2%	0.1%	66.31%

on directs that connect one BB terminal to another BB terminal, which is to be expected. Not only are BB terminals used for transfer, but also they tend to be the largest terminals in the network when measured by origin and destination freight volume. All direct types had an increase in pounds per trailer, but the EOL to EOL directs experienced the largest increase; the search appears to have removed low volume EOL-to-EOL direct moves from the load plan. The EOL to BB directs had the smallest increase in pounds per trailer, which is consistent with intuition because each EOL in the initial load plan only sends freight to a limited number of BB terminals, thus limiting opportunities for improvement. However, it seems that the local search technology is able to find changes to the load plan that substantially improve the trailer loading on final BB-to-EOL delivery legs.

It is also useful to measure the changes in a load plan by considering how many destination in-trees changed from the initial to final load plan; since the local search performs many iterations in the six hour run, and since each iteration may result in changes to the in-trees to many destinations simultaneously, we expect this approach to potentially find many changes. For this instance, 93.5% of the destination trees experienced some change from the initial load plan. While almost every destination in-tree was modified, only 3.5% of the commodities were given new transfer paths. Thus, a large portion of the in-trees (and the load plan) remained the same.

Given these details, it is clear that the large improvements found by our IP-based local search methodology are generated by increased consolidation identified during simultaneous evaluation of multiple destination trees. The generation of potential load plan paths during the search procedure, opposed to generating a set of potential load plan paths up front, also allows for more flexibility during the search routine accounting for some of the improvements. About 30% of the paths used in the improved load plan are paths that would not be considered when generating up to 50 shortest-path options for each commodity.

3.6.5 Comparison to In-tree IP-based Neighborhood Search

As a benchmark for the success of our IP-based neighborhood search technique, we used an operational version of the IP-based neighborhood search approach proposed by Erera et al. [19] as a comparison for cost improvements. In their research, an alternative neighborhood was used that fixed all freight paths to all but a single destination d in each iteration; a restricted version of $TSLP - IP$ (including the trailer balance constraints) is solved to search this neighborhood directly. Additionally, all freight paths used in the heuristic were generated *a priori* using a k-shortest paths approach. For more details on this approach, see Erera et al. [19].

Table 15: Comparison Between Single In-tree Neighborhood Search and NewOrOldTree Neighborhood Search

		1 Hour Runs				*6 Hour Runs*			
		Intree	*NewOrOldTree*			*In-tree*	*NewOrOldTree*		
		-	RO	AO	Rand_0.75	-	RO	AO	Rand_0.25
Nov10-W4	$\Delta Cost$	1.14	3.15	3.88	4.10	3.62	6.75	6.46	7.16
	ΔPPT	0.28	4.62	5.77	5.68	2.18	11.40	12.22	13.23
Dec10-W1	$\Delta Cost$	1.46	2.85	3.98	4.08	4.19	6.47	6.29	6.90
	ΔPPT	0.28	4.48	6.00	6.19	2.23	11.35	11.66	13.22
Dec10-W2	$\Delta Cost$	0.00	2.84	4.08	3.89	4.01	6.74	6.33	7.08
	ΔPPT	0.17	4.17	5.99	5.82	2.49	11.70	12.76	13.21
Feb11-W3	$\Delta Cost$	1.46	3.22	4.54	4.00	2.28	6.84	6.88	7.20
	ΔPPT	0.12	4.81	6.42	5.55	1.44	11.74	13.23	13.61
Mar11-W4	$\Delta Cost$	0.00	3.30	3.76	4.01	3.70	6.52	6.24	6.11
	ΔPPT	0.15	4.30	5.06	5.72	1.61	11.37	11.64	10.65
May11-W1	$\Delta Cost$	1.04	2.73	4.03	3.94	2.75	6.50	6.31	6.78
	ΔPPT	0.00	4.02	5.92	5.41	0.78	11.59	11.56	12.34

Table 15 presents results comparing the percentage cost improvements ($\Delta Cost$) and increase in pounds per trailer (ΔPPT) for both one hour and six hour runs of the *In-tree* neighborhood search and the *NewOrOldTree* neighborhood search. For the *NewOrOldTree* search, the reduce only (RO), attract only (AO), and the best performing mixed approach for each time limit are presented. In all instances, the *NewOrOldTree* search outperforms the *In-tree* search with respect to both the cost savings and pounds per trailer metrics. Also, the *NewOrOldTree* search achieves comparable or greater savings in the one hour runs when compared to the *In-tree* search in the six hour runs. Therefore, our approach not only generates greater savings during the six hour time period, but can generate similar cost savings to a longer *In-tree* search run in a shorter period of time if necessary.

3.7 Lower Bounds

In the computational results portion of this chapter, we measured the quality of a load plan by calculating the cost savings found with respect to the load plan that is currently being operated in practice. An alternative way of measuring the quality of a solution is to compare the load plan to some known lower bound. Because the load plan design literature has given little attention to the evelopment of lower bounds, we discuss in this section some initial ideas to create lower bounds.

In Section 3.4, the load plan design problem was modeled using a path-based formulation. Formulating the problem this way provided some benefits, like limiting the number of potential paths, ensuring a freight commodity is not served by more than one path, and potentially creating smaller formulations in terms of the number of constraints and variables. But, the lower bound created by the path-based formulation is not a true lower bound for the problem since it only holds for the paths used in the formulation. For the IP-based local search solution approach in this chapter, it is not possible to use the path-based formulation to create a lower bound since formulating the instances considered with all of the paths that can be used by our approach is not tractable. Therefore, it is necessary to consider alternative formulations and approaches to create lower bounds.

The load plan design problem can also be formulated using an arc-based formulation. For this formulation, we now introduce some additional notation to compliment the model presented earlier for the path-based formulation. The set of commodities is denoted by K, but now commodities are defined as all inbound freight to a single destination node $v \in N$. Let n_v^k represent the node supply for commodity k at node v. This value will be equal to the weight in pounds at the origin nodes for commodity k, the negative sum of all inbound weight at the destination for commodity k, and 0 at every other node. Let D_d represent the sum of all the freight destined for destination d, thus $D_d = n_v^k$ when $d = v = d(k)$.

In this formulation, we no longer have path variables, x_p^k. Instead, freight flows are represented with arc-based flow variables f. The f variables represent how much freight for commodity k uses arc a, *i.e.*, $f_a^k \in \Re \;\; \forall k \in K, \;\; \forall a \in A$. Second, y variables enforce consistency between paths for commodities heading to common destinations by indicating whether direct $l \in \Delta^+(u)$ is chosen for all commodities destined for terminal d routed through terminal u, *i.e.*, $y_l^d \in \{0,1\} \;\; \forall d \in U, \;\; \forall l \in \Delta^+(u), u \in U$. Finally, τ variables count the number of trailers (empty or loaded) that move on arc a, i.e., $\tau_a \in \mathbb{Z}_+ \;\; \forall a \in A$.

The formulation is to then minimize

$$\sum_{a \in A} c_a \tau_a + \sum_{k \in K} \sum_{a \in A} h_{o(a)}(f_a^k - n_{o(a)}^k)$$

subject to

$$\sum_{a \in \delta^+(v)} f_a^k - \sum_{a \in \delta^-(v)} f_a^k = n_v^k \;\; \forall v \in N, k \in K \tag{18}$$

$$\sum_{l \in \Delta^+(u)} y_l^d \leq 1 \;\; \forall u \in U, \;\; \forall d \in U \tag{19}$$

$$f_a^k \leq D_d \, y_{l(a)}^{d(k)} \;\; \forall k \in K, \forall a \in A \tag{20}$$

$$\sum_{k \in K} f_a^k \leq \tau_a \;\; \forall a \in A \tag{21}$$

$$\sum_{a \in \delta^+(v)} \tau_a - \sum_{a \in \delta^-(v)} \tau_a = 0 \;\; \forall v \in N \tag{22}$$

The objective is to minimize the total transportation and handling costs. Constraints (18) are freight flow balance constraints and ensure that the freight is moved from its origin to destination for each commodity. Constraints (19) ensure that a single outbound direct is selected for each terminal u and freight destined for terminal d. Constraints (20) ensure that freight flow for each commodity k is only allowed on directs that are chosen. Constraints (21) ensure that there are enough trailers moved along an arc to carry the freight assigned to the arc via the paths chosen. Finally, constraints (22) ensure flow balance of trailers at every node in the time-space network, and thus ensure proper repositioning of trailers.

3.7.1 Comparing Load Plan Design Formulations

To help understand the differences of an arc-based formulation versus path-based formulation, we performed preliminary experimentation on a simple 10 terminal network with three varying sets of freight flows. The **1day-10**, **2day-10**, and **5day-10** instances used for this experimentation see freight demands requested on one, two, or five days a week, respectively, for each o-d pair in the network. The total weekly freight demand between each o-d pair is the same for all three of the instances considered. Table 16 contains the results for the three instances considering the use of 1, 2, or 3 BB terminals. In this table, the number constraints, number of variables, solve time in seconds, objective function, lower bound, and optimality gap is provided for both the path-based formulation and arc-based formulation. Each formulation was given six hours to solve.

The information in Table 16 shows that the arc-based formulations have a significantly larger number of constraints, while there are a larger number of variables in the path-based formulations. These differences in the formulations arise because of the nature of the modeling approach used in each of them. In the path-based formulations, the number of variables and constraints increases as the number of freight commodities and BB terminals increases. The larger number of constraints in the arc-based formulations results from needing to include a copy of the freight balance constraints at every node for every destination node in the time-space terminal that has freight destined to it.

The path-based formulations solve faster in most of the cases tested, with the exception of the **5day-10** instance considering 2 or 3 BB terminals that use the entire six hour time allotment. The arc-based formulation for all instances considered took the entire six hour time limit and all of the optimality gaps were larger than 13% when the time limit was reached. The faster solution time for the path-based formulations can be largely attributed to the limited set of timed paths considered, while the arc-based formulations consider all possible timed path combinations. The path-based formulation also only considers paths with up to two intermediate stops. If one or two BB terminals are used, this is a complete enumeration of the static path options, but when three BB terminals are considered this does not create all feasible static paths. Because of the path generation, the lower bound created by the path-based formulation is not a true lower bound for the problem. The **2day-10** instance considering 1 or 2 BB terminals provides an example where the path-based formulation does not provide a valid lower bound for the load plan design problem.

Table 16: Comparison of Path-Based Formulation versus Arc-Based Formulation

		Path-Based Formulation						Arc-Based Formulation					
Instance	# BBs	# Constraints	# Variables	Solve Time (sec)	Objective Function	Lower Bound	Optimality Gap	# Constraints	# Variables	Solve Time (sec)	Objective Function	Lower Bound	Optimality Gap
1day-10	1	2048	8446	0.56	87030	87030	0.00%	324579	7864	22272.3	93709.1	80734.2	13.85%
1day-10	2	3456	8896	2.59	85721.1	85721.1	0.00%	324579	7864	22416.9	92268.9	79256.3	14.10%
1day-10	3	4105	9082	3.31	84856.1	84855.3	0.00%	324579	7864	22095.6	90220.7	78074.7	13.46%
2day-10	1	3237	9028	2.91	97817.2	97817.2	0.00%	359849	7864	22155.8	97318.8	78757.4	19.07%
2day-10	2	5892	9919	68.37	95775.2	95775.2	0.00%	359849	7864	22290.9	95124	78060.3	17.94%
2day-10	3	7127	10285	369.38	93536.3	93535.4	0.00%	359849	7864	22188.8	97934.6	77935.9	20.42%
5day-10	1	6713	10774	18.1	105260	105259	0.00%	359849	7864	21992.2	111294	78452.4	29.51%
5day-10	2	13078	13015	21671.1	100221	98613.3	1.63%	359849	7864	21674.6	147692	77956.7	47.22%
5day-10	3	16038	13939	21665.6	94115.3	90422.4	4.08%	359849	7864	21676.5	178823	77874.9	56.45%

In these two cases, the solution to the arc-based formulation has a lower total operating cost than the solution found by the path-based formulation, even though the optimality gap for the formulation is still larger than 17%.

3.7.2 Static Load Plan Design Formulations

Because the load design formulations using the timed linehaul network $TS - LN = (N, A)$ incorporating freight dispatch timing can still be difficult to solve for a small 10 terminal network, we were interested in lower bounds that could be created by modifying the load plan design formulations to ignore freight dispatch timing and use the static linehaul network $LN = (U, L)$. To make static versions of the path-based and arc-based formulations, modifications need to be made so that freight and trailer flows are modeled on the static linehaul network using the directs and terminals instead of using the timed linehaul network with arcs and nodes. This means that trailers are now counted on directs, which only represent moving between two terminal locations and do not consider the day or time during the week. Also, all trailer or freight balance equations are created for each terminal, rather than each node. When the load plan design problem is modeled on a static network, the weekly aggregate freight flows for each o-d pair are considered commodities instead of the individual freight demands between each o-d pair.

The static version of the arc-based load plan design formulation does not enforce service feasibility since there is no consideration of timing when using the static linehaul network. Therefore, the freight flows defined by the solution to the formulation may not result in a feasible solution to the timed version of the problem. Therefore, the solution to the static arc-based formulation provides a lower bound because there may be (1) time infeasible paths, (2) time infeasible consolidation, and (3) freight sent along more than one path.

Table 17 contains results comparing the solutions for the timed and static versions of the load plan design formulations. First, the path-based formulations are compared for the three instances considering the use of 1, 2, or 3 BB terminals, followed by a comparison of the arc-based formulations for the same scenarios. For each of the formulations, the number of constraints, number of variables, solve time ins seconds, objective function, lower bound,

Table 17: Comparison of Timed versus Static Formulations

		Path-Based Formulation						Static Path-Based Formulation					
Instance	# BBs	# Constraints	# Variables	Solve Time (sec)	Objective Function	Lower Bound	Optimality Gap	# Constraints	# Variables	Solve Time (sec)	Objective Function	Lower Bound	Optimality Gap
1day-10	1	2048	8446	0.56	87030	87030	0.00%	478	1096	0.03	79216.1	79216.1	0.00%
1day-10	2	3456	8896	2.59	85721.1	85721.1	0.00%	796	1221	1.63	76952.8	76952.8	0.00%
1day-10	3	4105	9082	3.31	84856.1	84855.3	0.00%	981	1283	5.3	75945.4	75945	0.00%
2day-10	1	3237	9028	2.91	97817.2	97817.2	0.00%	478	1096	0.02	79216	79216	0.00%
2day-10	2	5892	9919	68.37	95775.2	95775.2	0.00%	796	1221	2.03	76764.3	76763.6	0.00%
2day-10	3	7127	10285	369.38	93536.3	93535.4	0.00%	981	1283	13.48	75968.8	75968.1	0.00%
5day-10	1	6713	10774	18.1	105260	105259	0.00%	478	1096	0.03	79215.9	79215.9	0.00%
5day-10	2	13078	13015	21671.1	100221	98613.3	1.63%	796	1221	2.23	76764.1	76764.1	0.00%
5day-10	3	16038	13939	21665.6	94115.3	90422.4	4.08%	981	1283	15.21	75945.1	75944.4	0.00%
		Arc-Based Formulation						Static Arc-Based Formulation					
Instance	# BBs	# Constraints	# Variables	Solve Time (sec)	Objective Function	Lower Bound	Optimality Gap	# Constraints	# Variables	Solve Time (sec)	Objective Function	Lower Bound	Optimality Gap
1day-10	1	324579	7864	22272.3	93709.1	80734.2	13.85%	1788	954	4852.57	79146.1	79145.3	0.00%
1day-10	2	324579	7864	22416.9	92268.9	79256.3	14.10%	1788	954	21777.5	76672.3	75532.8	1.49%
1day-10	3	324579	7864	22095.6	90220.7	78074.7	13.46%	1788	954	21762.9	75963.2	73393.4	3.38%
2day-10	1	359849	7864	22155.8	97318.8	78757.4	19.07%	1788	954	3612.96	79146	79145.2	0.00%
2day-10	2	359849	7864	22290.9	95124	78060.3	17.94%	1788	954	21797.4	76672.2	75257.7	1.84%
2day-10	3	359849	7864	22188.8	97934.6	77935.9	20.42%	1788	954	21761.1	75809.9	73383.2	3.20%
5day-10	1	359849	7864	21992.2	111294	78452.4	29.51%	1788	954	2303.63	79145.9	79145.1	0.00%
5day-10	2	359849	7864	21674.6	147692	77956.7	47.22%	1788	954	21954.3	76672	75534.8	1.48%
5day-10	3	359849	7864	21676.5	178823	77874.9	56.45%	1788	954	21774.8	75687.2	73455.1	2.95%

and optimality gap are reported.

The results in Table 17 show that the static path-based formulations are able to solve to optimality in seconds for all instances considered, which can be attributed to the much smaller formulation in terms of variables and constraints when modeling the problem on a static network. Like in the static path-based formulation, the number of constraints and variable is notably smaller for the static arc-based formulation. The static arc-based formulations solve to optimality for the instances that consider using 1 BB terminal, while the remaining instances use the full six hours. Even though some of the instances still take six hours, the optimality gaps are much smaller, with all of them being less than 4%. However, the lower bound values for the timed formulations are stronger, indicating that if a solution time of six hours is needed, then solving the timed version of the arc-based formulation provides more value.

For both static formulations, the total operating costs are much smaller than those observed in the solutions for the timed versions of the formulations since the weekly aggregate freight flows are used instead of daily freight flows. It should be noted that the some of the solutions from the static arc-based formulations have smaller operating costs than the path-based formulations. This is due to the fact that the static arc-based formulation does not enforce service feasibility and allows freight to be split on more than one path.

3.7.3 Linear Programming Relaxations for Arc-Based Formulations

Since both the timed and static versions of the arc-based load plan design formulations can be difficult to solve on a small 10 terminal network, there is some interest in understanding the strength of the LP-relaxation since it solves faster. If there are only small improvements identified for the lower bound during the six hour solution time, then solving the LP-relaxation would provide a relatively strong bound in a more reasonable amount of time. Table 18 contains results from solving the LP-relaxation for both the timed and static version of the arc-based formulation. In this table, the LP-relaxation solve time in seconds and objective function and IP solve time in seconds and lower bound are reported for both formulations. For the timed arc-based formulation, there is only a small amount of improvement for the lower bound identified during the six hour solve time when compared to the lower bound provided by the LP-relaxation that can be found in less than a minute for all instances. The static arc-based formulation is able to make more notable improvements on the lower bound during the six hour time period. Even though the LP-relaxation can be solved in less than a second for all instances, this lower bound is not as strong as the one produced by LP-relaxation for the timed arc-based formulation. Also, some of the improved lower bounds found during the six hour time period for the static arc-based formulations are still weaker than the LP-relaxation of the timed arc-based formulation.

Table 18: Arc-Based Formulations LP-Relaxations

	Arc-Based Formulation				Static Arc-Based Formulation			
	LP-relaxation		IP Solve		LP-relaxation		IP Solve	
Instance, # BBs	Solve Time	Objective Function	Solve Time	Lower Bound	Solve Time	Objective Function	Solve Time	Lower Bound
1day-10, 1	5.41	78148.4	22272.3	80734.2	0.01	66366.4	4852.57	79145.3
1day-10, 2	7.46	77944.6	22416.9	79256.3	0.01	66151.9	21777.5	75532.8
1day-10, 3	10.34	77867.5	22095.6	78074.7	0.01	66074.8	21762.9	73393.4
2day-10, 1	7.25	78147.2	22155.8	78757.4	0.01	66365.4	3612.96	79145.2
2day-10, 2	9.1	77943.5	22290.9	78060.3	0.01	66150.9	21797.4	75257.7
2day-10, 3	27.73	77866.4	22188.8	77935.9	0.02	66073.8	21761.1	73383.2
5day-10, 1	12.46	78144	21992.2	78452.4	0.01	66362.6	2303.63	79145.1
5day-10, 2	18.17	77940.3	21674.6	77956.7	0	66148.2	21954.3	75534.8
5day-10, 3	32.86	77863.2	21676.5	77874.9	0.01	66071.1	21774.8	73455.1

3.7.4 Static Arc-Based Formulation Variations

Even though the timed arc-based formulation provides stronger lower bounds than the static arc-based formulation, there is value in exploring more bounds related to the static arc-based formulation since it is a much smaller formluation with respect to the number of constraints and variables. Therefore, lower bounds generated using some variation of the static arc-based formulation may scale better for larger networks than the bounds using the time arc-based formulation. Another way to find a valid lower bound without having to solve the arc-based formulation for the six hour allotted time period is to relax some aspects of the load plan design problem. To see if solving a relaxed version of the problem would allow a stronger lower bound to be generated in a shorter period of time, we considered two relaxations. The first relaxation ignores the intree property used when routing freight, which is referred to as the "Static Arc-Based, No Intree" formulation. This formulation removes constraints (19) and (20) from the static version of the arc-based formulation. The second relaxation modifies the trailer transportation costs and removes empty repositioning, which is referred to as the "Static Arc-Based, No Trailers" formulation. This formulation modifies the set of constraints (21) to be an equality constraint and removes constraints (22) from the static version of the arc-based formulation. Also, the trailer variables are changed from integer variables to continuous variables, so that this variable is equal to the sum of all freight traveling on a single direct.

Table 19: Static Arc-Based Formulation Variations

	Static Arc-Based, No Intree				Static Arc-Based, No Trailers			
Instance, # BBs	# Constr	# Vars	Solve Time	Lower Bound	# Constr	# Vars	Solve Time	Lower Bound
1day-10, 1	258	144	492.36	68010.6	1778	810	0.02	60541.2
1day-10, 2	258	144	468.28	68010.6	1778	810	0.03	60326.7
1day-10, 3	258	144	430.31	68010.6	1778	810	0.05	60248.8
2day-10, 1	258	144	347.45	68000	1778	810	0.02	60540.2
2day-10, 2	258	144	302.49	68000	1778	810	0.05	60325.7
2day-10, 3	258	144	359.48	68000	1778	810	0.04	60247.8
5day-10, 1	258	144	21.59	67931.7	1778	810	0.02	60537.3
5day-10, 2	258	144	27.24	67931.7	1778	810	0.03	60322.9
5day-10, 3	258	144	20.43	67931.7	1778	810	0.04	60245

Table 19 reports the results from solving the restricted version of the static arc-based formulation. This table provides the number of constraints, number of variables, solve time in seconds, and lower bound produced by each of the alternative formulations considered. The "Static Arc-Based, No Intree" formulations solve in less than ten minutes for all of the instances considered in this study. The solutions from this restricted version of the static arc-based formulation result in a stronger bound than just solving the LP-relaxation for the static arc-based formulation. The "Static Arc-Based, No Trailers" formulations solve in less than a second for all instances, but produce the weakest lower bound of all the approaches considered.

The "Static Arc-Based, No Trailers" formulation produces the weakest lower bound because it identifies a solution that sends all commodities directly from origin to destination as long as the network considered is a complete network. Because the empty repositioning constraints are removed from this formulation, the resulting bound is weaker than simply solving the LP-relaxation of the static arc-based formulation. For the static arc-based formulation, the trailer variables drive more freight routing decisions than the intree varialbes and corresponding constraints, which can be seen by the stronger lower bound produced by the "Static Arc-Based, No Intree" formulation.

3.7.5 Lower Bound Summary

In this section, we have presented several possible directions for developing a lower bound for the load plan design problem. The focus of this discussion has centered primarily on lower bounds that can be created using the arc-based load plan design formulation as a starting point. Table 20 summarizes the lower bounds presented using variations of the arc-based formulation. This table reports the solve time in seconds and lower bound generated by the timed arc-based LP-relaxation, the "Static Arc-Based, No Intree" formulation, the static arc-based LP-relaxation, and the "Static Arc-Based, No Trailers" formulation. The results clearly show that the strongest lower bound is found by solving the LP-relaxation of the timed arc-based formulation. Because we only tested this approach on a small 10 terminal network, it is not clear that this bound can be computed efficiently for large networks, like the ones seen in practice by a large LTL carrier. Therefore, the testing of these approaches on larger networks should be noted as an area of future work.

Table 20: Arc-Based Lower Bound Summary

	Arc-Based LP-Relaxation		Static Arc-Based, No Intree		Static Arc-Based LP-Relaxation		Static Arc-Based, No Trailers	
Instance, # BBs	Solve Time	Lower Bound	Solve Time	Lower Bound	Solve Time	Lower Bound	Solve Time	Lower Bound
1day-10, 1	5.41	78148.4	492.36	68010.6	0.01	66366.4	0.02	60541.2
1day-10, 2	7.46	77944.6	468.28	68010.6	0.01	66151.9	0.03	60326.7
1day-10, 3	10.34	77867.5	430.31	68010.6	0.01	66074.8	0.05	60248.8
2day-10, 1	7.25	78147.2	347.45	68000	0.01	66365.4	0.02	60540.2
2day-10, 2	9.1	77943.5	302.49	68000	0.01	66150.9	0.05	60325.7
2day-10, 3	27.73	77866.4	359.48	68000	0.02	66073.8	0.04	60247.8
5day-10, 1	12.46	78144	21.59	67931.7	0.01	66362.6	0.02	60537.3
5day-10, 2	18.17	77940.3	27.24	67931.7	0	66148.2	0.03	60322.9
5day-10, 3	32.86	77863.2	20.43	67931.7	0.01	66071.1	0.04	60245

Now, using the lower bound generated by solving the timed arc-based LP-relaxation, we can demonstrate the alternative ways of measuring the quality of a solution. Recall that in this chapter, we have used a cost savings measure with respect to a currently operated load plan as a way measuring the quality of a solution. Given a valid lower bound, we can express the quality of a solution using a relative optimality gap as $\frac{|BS-LB|}{|LB|}$, where BS represents the best solution found and LB represents the best lower bound found.

Table 21: Measuring the Quality of a Solution

Instance	# BBs	$\Delta Cost$	Relative Optimality Gap	
			Initial Solution	Improved Solution
1day-10	1	6.9%	22.7%	14.3%
1day-10	2	7.9%	23.1%	13.4%
1day-10	3	9.4%	23.2%	11.7%
2day-10	1	22.1%	69.3%	31.9%
2day-10	2	29.1%	69.7%	20.3%
2day-10	3	29.9%	69.9%	19.1%
5day-10	1	48.2%	253.3%	83.0%
5day-10	2	59.7%	254.2%	42.7%
5day-10	3	60.6%	254.6%	39.8%

Table 21 contains results for the three instances tested in this section comparing the alternative methods for measuring the quality of a solution. Because these instances were generated, the initial solution used for comparison is a direct load plan that routes each freight commodity directly from its origin to destination without using any intermediate transfers. The load plan neighborhood search approach presented in this chapter was used to generate the improved load plan solution. Given the size of the network, we assigned a 30 minute time limit for identifying an improved solution.

The results in Table 21 depict the differences in the two methods used for measuring the quality of a solution. The cost savings measurement ($\Delta Cost$) provides a comparison with an existing solution, whereas the relative optimality gap provides a comparison with a lower bound. Because the lower bound may not be strong, the relative optimality gap for a good solution can still be very large. But, by considering the improvement in the relative optimality gap from the initial solution to the improved solution, more information about the quality of the solution can be seen. For example, the fact that the initial relative optimality gap for the **5day-10** instance considering 3 BB terminals is 254.6% and the improved solution reduces the relative optimality gap to 39.8% holds more meaning than just knowing that there is 60.6% cost savings generated. Both forms of measurement provide interesting information, but the relative optimality gap also gives an indication of the overall quality of the solution with respect to potential opportunity for improvement.

CHAPTER IV

HUB LOCATION FOR LTL NETWORK DESIGN

4.1 Introduction

In the final chapter of this thesis, the strategic planning decision of determining the operating roles for a given set of terminals used to route freight through a LTL network is considered. When determining the operating roles for the set of terminals, every terminal must be operated so that the pickup and delivery operations do not change since these costs are independent of the terminal roles and not evaluated in this study. Each terminal in the network operates as either a BB or EOL, determining whether freight can be transferred at the facility to increase consolidation. Although this is a strategic planning problem, the operating role of a terminal can be changed at any point in time with relatively little fixed cost and resources. Since freight spends very little time in a BB terminal, additional space may not even be required to upgrade a terminal from an EOL to a BB. However, in the trucking industry, space and labor are easy to acquire if additional resources are necessary. For example, if additional space is required to support freight transferring activities when upgrading a terminal from an EOL to a BB terminal, it can be obtained by leasing a larger facility. Also, in many facilities, drivers participate in much of the freight loading/unloading activity for trailers, so additional labor can be easily acquired.

In the LTL industry, many companies expand their service area by acquiring smaller, regional freight carriers through buyouts or mergers. Under these circumstances, it is beneficial for a company to reevaluate the terminal operating roles so that redundancies in the new terminal network can be removed. Although this is a strategic planning problem that only arises periodically, the terminal role decisions greatly impact the daily freight routing operations. Choosing a sub-optimal set of BB terminals may result in significantly higher total operating costs. Therefore, it is important to investigate if and how freight dispatch times, origin-destination freight flows, and accurate operating costs impact the set of BB facilities sekected to operate in a given network. Techniques developed to determine an appropriate set of BB terminals would also be useful when freight flow patterns change significantly. The set of terminals in the network may not change, but the demands on the network may shift because of changes in customer needs, justifying the need to reevaluate terminal roles. In this research, we examine how the consideration of freight dispatch timing and trailer transportation costs impact the choices of BB terminals and the resulting load plan.

The remainder of this chapter is organized as follows. Section 4.2 provides a brief literature review of related hub location literature. Sections 4.3 and 4.4 formally define the problem and detail the proposed solution approach. Section 4.5 details additional formulations used as comparison for our model. Finally, Section 4.6 provides results from a computational study conducted for the problem.

4.2 Literature Review

The primary focus of this chapter is to determine which subset of terminals should operate as BB terminals given an existing terminal network, which can be classified as a hub location problem. The hub location problem has been extensively studied, with many variants appearing in literature beginning in the late 1980's. A survey of existing literature focusing on network hub location problems is provided by Alumur and Kara [2]. Their suvey discusses models classified under the p-hub median problem, the hub location problem with fixed costs, the p-hub center problem, and the hub covering problem. Within each of these problem variants, there is also a subdivision of single and multiple allocations, which refers to the assignment of non-hub terminals to hub locations. Since all of the classical hub location problems presented in their review are NP-hard (except some special cases), many of the papers discussed develop heuristic approaches to address the problems, including tabu search heuristics, simulated annealing heuristics, tailored branch-and-bound algorithms, and Lagrangia relaxation methods. The authors also discuss areas of the existing literature where there are deficiencies and propose topics for future research. They mention that the existing literature lacks realistic studies driven by problems arising in practice.

Timing is one aspect of real world problems that has been given little attention in the existing hub location literature. Freight dispatch timing is an increasingly important problem feature due to shorter service times and service standards that many companies are implementing to gain a competitive advantage. The way in which timing is considered in the existing literature varies greatly depending on the paper. Some literature considers timing as a mechanism for limiting transferring freight at a hub terminal ([17]), while other work considers timing as a measure of service ([10], [31], and [53]). Da Graca Costa, *et al.* [17] presented a different approach to the capacitated single allocation hub location problem using timing. In their problem, a second objective was introduced that tries to minimize the time to process flow entering the hubs instead of using capacity constraints to limit the amount of flow that can be received by the hubs. Their work presented two bi-criteria models. In the first model, total time is considered as the second criteria, and in the second model, the maximum service time for the hubs is minimized. A bi-criteria approach was developed to generate non-dominated solutions using an interactive decision-aid approach developed for bi-criteria integer programming problems. Computational results on both bi-criteria models demonstrated the value of the increased information provided by non-dominated solutions of the bi-criteria model when compared to the unique solutions given by the capacitated hub location model.

Campbell [10] introduced cost minimizing hub location models for time definite transportation, where each origin-destination pair has a specified service level that must be met. The service levels are imposed by limiting the maximum travel distance via the hub network for each origin-destination pair. Models were proposed introducing time definite transportation in multiple allocation p-hub median problems and hub arc location models. Computational results solving the models using CPLEX showed that an increase in the service level may require modifying a given hub network by relocating hubs or by adding hubs and hub arcs. These changes may increase the fixed costs for the new facilities and assets, while decreasing transportation costs.

Work in Kara and Tansel [31] and Yaman, *et al.* [53] incorporated the transient time spent at hubs for unloading, loading, and sorting operations, which may constitute a significant portion of the total delivery time in cargo systems. The focus of these papers is on the minimization of the last arrival time at destinations in cargo delivery systems, which is named the Latest Arrival Hub Location Problem. Kara and Tansel [31] developed a model that correctly computes arrival times by taking into account both the freight times and transient times. Computational results using standard data sets demonstrate that medium-sized problems ($n = 25$) can effectively be solved using commercial optimization solvers. Yaman, *et al.* [53] extend the problem beyond the standard location and allocation decisions to address issues relating to multiple stopovers, design of vehicle routes, and waiting times at hubs. Their problem variant is named the Latest Arrival Hub Network Design Problem. They propose a general model incorporating the main features of ground transportation delivery systems and provide valid inequalities and lifting results to strengthen the model. Computational studies on the standard CAB data set demonstrate the use of the inequalities in solving the problem. The model was also tested with a data set constructed from locations in Turkey.

The literature most closely related to the problem addressed in this chapter focuses on defining hub locations in LTL networks. In this area, work has been done by Cunha and Silva [16] and the previously discussed work by Yaman, *et al.* [53]. Cunha and Silva [16] addressed a modified version of the Uncapacitated Hub Location Problem with Single Allocation (UHP-S) in which the discount factor on the hub-to-hub directs varies according to the total freight between the hub terminals. This variation was motivated by the desire to configure a hub-and-spoke network for the operations of a LTL carrier in Brazil. To solve the UHP-S variant , they proposed a genetic algorithm to find the number of hubs, location of the hubs, and the assignment of spokes to hubs. The local search heuristic is based on a simplified simulated annealing mechanism that allows non-improving solutions to be accepted with a certain probability. The computational results demonstrate the efficiency and effectiveness of the proposed approach.

To the best of our knowledge, there is no literature considering timing of freight dispatches representing daily operations and consolidation opportunities. All existing timing considerations focus on the impact of time on service standards and ignore the impact on transportation costs and routing decisions. Also, there has not been any work explicitly considering trailer transportation costs on directs offering service. All existing literature uses a discount factor to represent the economies-of-scale achieved when freight flow increases on hub-to-hub directs. The existing hub location approaches use relatively simple models of cost and time. This research investigates models with more accurate representations of cost and time, which involve the (1) investigation of the benefits and (2) investigations of how to solve these more complex models.

4.3 Hub Location Load Plan Design Integer Program

In this work, we introduce a new variant of the hub location problem that incorporates trailer transportation costs using a time-space network to accurately capture consolidation opportunities. The goal of this research is to (1) gain insight on how accurate, detailed modeling impacts the choice of BB terminals and (2) develop appropriate solution approaches for the Hub Location Load Plan Design (HLTLD) problem. The Hub Location Load Plan Design problem determines the subset of terminals to be operated as BB terminals, in addition to defining the corresponding load plan given the set of selected BBs. Let us define the notation for the HLTLD integer programming formulation.

Given networks $LN = (U, L)$ and $TS - LN = (N, A)$, let $\Delta^+(u) \subseteq L$ denote the set of potential outbound directs from terminal $u \in U$, let $l(a)$ denote the direct $l \in L$ corresponding to the arc $a \in A$, let $d(l)$ denote the destination of direct $l \in L$, let $\delta^+(n) \subseteq A$ denote the set of outbound arcs from node $n \in N$, let $\delta^-(n) \subseteq A$ denote the set of inbound arcs to node $n \in N$, and let c_a denote the per-trailer travel cost along arc $a \in A$. The set of commodities is denoted by K. For each commodity $k \in K$, $o(k) \in U$ is the origin terminal, $d(k) \in U$ is the destination terminal, w_k is the weight in pounds, and q_k is the size measured in fractional trailers (note, q_k need not be less than one). The set of paths that serve commodity $k \in K$ is denoted $P(k)$. Let BB_{max} be the maximum number of BB terminals that can be selected and let $B \subseteq U$ be the set of terminals considered as BB candidates. A maximum number of BB terminals is defined so that every terminal does not allow freight transfer activity. In practice, approximately 20% of the terminals are operated as BB terminals, so we include a constraint on the maximum allowing planners to control the ratio of BB terminals to all terminals. Restricting the number of transfer terminals could be incorporated into the objective function instead of using a constraint, but representing the cost trade-offs between using a new BB terminal and routing freight through an alternative terminal is complicated. Therefore, we chose to restrict the number of BB terminals using a constraint since we can use industry practices as a guideline for establishing an acceptable number of BB terminals.

The integer program has four sets of decision variables. First, z variables are binary and have a value of one if terminal d is selected to be a BB and zero otherwise, *i.e.*, $z_d \in \{0,1\} \ \forall d \in B$. Second, x variables are binary and have a value of one if commodity k uses path p and zero otherwise, *i.e.*, $x_p^k \in \{0,1\} \ \ \forall k \in K, \ \ \forall p \in P(k)$. Next, y variables are binary and enforce consistency between paths for commodities heading to common destinations by having a value of one if direct $l \in \Delta^+(u)$ is chosen for all commodities destined for terminal d routed through terminal u and zero otherwise, *i.e.*, $y_l^d \in \{0,1\} \ \ \forall d \in U, \ \ \forall l \in \Delta^+(u), u \in U$. Finally, τ variables count the number of trailers (empty or loaded) that move on arc a, i.e., $\tau_a \in \mathbb{Z}_+ \ \ \forall a \in A$.

The formulation is to then minimize

$$\sum_{a \in A} c_a \tau_a + \sum_{k \in K} \sum_{p \in P(k)} h_p w_k x_p^k$$

subject to

$$\sum_{p \in P(k)} x_p^k = 1 \ \ \forall k \in K \tag{23}$$

$$\sum_{l \in \Delta^+(u)} y_l^d \leq 1 \ \ \forall u \in U, \ \ \forall d \in U \tag{24}$$

$$\sum_{p \in P(k): a \in p} x_p^k \leq y_{l(a)}^{d(k)} \ \ \forall k \in K, \forall a \in A \tag{25}$$

$$y_l^d \leq z_{d(l)} \ \ \forall d \in B, \forall l \in L : d(l) \neq d \tag{26}$$

$$\sum_{d \in B} z_d \leq BB_{max} \tag{27}$$

$$\sum_{k \in K} \sum_{p \in P(k): a \in p} q_k x_p^k \le \tau_a \;\; \forall a \in A \tag{28}$$

The objective is to minimize the total transportation and handling costs. Constraints (23) ensure that a single path is chosen for each commodity. Constraints (24) ensure that a single outbound direct is selected for each terminal u and freight destined for terminal d. Constraints (25) ensure that a path can only be chosen for commodity k when all of its component directs are chosen. Constraints (26) ensure that a direct can only be chosen as part of a directed intree to a destination terminal if the intermediate terminal is selected as a BB. Constraint (27) ensures that the set of BBs selected contains no more than the allowed number. Constraints (28) ensure that there are enough trailers moved along an arc to carry the freight assigned to the arc via the paths chosen. Note that the constraints used to ensure proper empty repositioning of trailers through a flow balance of trailers at every node in the time-space network have been taken out of this formulation. We have chosen to ignore empty repositioning in this problem setting since considering these constraints makes the size of the integer program larger.

Constraints (26) ensure that the intree decisions for each destination and path chosen for each of the commodities correspond to the set of terminals selected to operate as BB terminals. In the formulation presented, the directs used in the intree for each destination are limited by the selected set of BB terminals. This set of constraints can also be modeled by restricting the set of feasible paths for each commodity given a set of BB terminals. In this case, the constraints would be the following:

$$x_p^k \le z_d \;\; \forall k \in K, \forall d \in B, \forall p \in P(k) : d \in p, d \ne o(k), d \ne d(k). \tag{29}$$

This set of constraints ensures that a path can only be chosen when the intermediate terminals are selected to be BB terminals. Although modeled differenty, this achieves the same goal as the set of constraints (26). We chose to model this aspect of the prolem using constraints (26) because we achieved better quality computational results using this set of constraints.

4.3.1 Path Creation

All approaches in this chapter consider the same set of paths to route freight while identifying the set of terminals to operate as BB terminals. To mimic the behavior of the hub location formulation presented later in Section 4.5.1, only paths with up to two intermediate BB stops are considered. This means that direct paths between a commodity's origin and destination, one-stop paths visiting a single intermediate location, and two-stop paths visiting two intermediate locations are considered. Because many LTL networks operate a hub-and-spoke network where most of the shipments use two intermediate stops, considering paths with up to two intermediate stops is a sufficiently good approximation of the consolidation activities seen in practice. Additionally, only paths that are service feasible for a commodity are considered as potential paths. Service feasibility is verified by calculating the time needed to travel from a path's origin to the intermediate stop(s) to the final destination and confirming that this time does not exceed the given service standard for the commodity. This path set is denoted P and contains all feasible direct, one-stop, and two-stop paths for each commodity.

In the facility location formulation presented later in Section 4.5.2, a set of paths is not explicitly defined. However, the nature of the formulation restricts potential paths to the set of direct, one-stop, and two-stop paths that are considered by the other formulations presented.

4.4 IP-based Local Search Approach

Because the Hub Location Load Plan Design (HLTLD) formulation incorporates both the hub location and load planning aspects of the problem, this formulation can be difficult to solve even on small networks. To ensure that high-quality solutions are found reliably given a limited computation time, we developed an effective IP-based neighborhood search heuristic. This heuristic uses a combination of heuristic search and optimization to incrementally find improving solutions. By solving restricted versions of the HLTLD formulation, in addition to IP formulations tailored to search specific neighborhoods, the approach is able to find improving solutions more effectively than directly solving the HLTLD formulation using a commercial solver.

After an initial solution is generated, the integer program solved each iteration only uses a subset P' of the full path set P. The subset P' includes the set of paths that form the best known feasible solution to ensure that solving the restricted version of the HLTLD formulation will never result in an inferior solution. Thus, for each commodity, the optimization problem chooses between the path that serves that commodity in the best known solution and the alternative paths that are also available in P' to serve that commodity. Since even the restricted integer program defined over the path subset P' may be difficult to solve, we impose a computation time limit for each iteration. The goal is to have path restrictions P' in each iteration that result in improved feasible solutions quickly. (It is not important to prove optimality of an improved solution to the restricted problem solved.)

Our approach for restricting the path set P is to focus on freight flowing through specific BB terminals or varying combinations of terminals during each iteration. In the neighborhoods considered during each iteration, we only place restrictions on the path variables (x_p^k) for the set of paths P. All of the intree variables (y_l^d), trailer count variables (τ_a), or BB variables (z_d) are always included in the neighborhood. Recall that the set of terminals considered as BB candidates is denoted $B \subseteq U$. When defining the neighborhoods, we will refer to the set of BB terminals in the best known feasible solution as B'. We consider the following restriction neighborhoods that add additional paths to P' to supplement the paths that form the current best known feasible solution:

1. **Reduce Freight at BB**

   Let $K' \subset K$ be the set of commodities that are currently served by a path visiting a specific intermediate transfer terminal $b \in B'$. The restricted path set P' contains all paths that serve only commodities in K' and visit only alternative transfer terminals $b \in B' \setminus b$.

2. **Attract Freight to New BB**

   Let $K' \subset K$ be the set of commodities that currently use a direct path from $o(k)$ to

$d(k)$ to serve the commodity. The restricted set P' contains all paths that serve only commodities in K' and visit at least one new transfer terminal $b \in B \setminus B'$.

3. **Attract Freight to BB**

   For this restriction, we consider all commodities K. The restricted path set P' contains all paths that visit a specific BB terminal $b \in B'$.

4. **Attract Freight to Any BB**

   For this restriction, we consider all commodities K. The restricted path set P' contains all paths that visit intermediate transfer terminals $b \in B'$.

5. **Swap BB**

   For this restriction, we consider all commodities K. Let $M \subseteq B \setminus B'$ be a set of neighboring terminals to a specific BB terminal $b \in B'$. These neighboring terminals are chosen to be the five terminals closest to b that are not current BB terminals. The restricted path set P' contains all paths that visit only intermediate transfer terminals $b \in B' \cup M$.

After initial experimentation, the **Attract Freight to Any BB**, **Attract Freight to BB**, and **Reduce Freight at BB** neighborhods were found to only make improvements on the intree decisions for each destination and path choices for each commodity. Since these neighborhoods made no changes to the set of terminals selected to be BB terminals, we replaced these neighborhoods with the Load Plan Neighborhood Search presented in the previous chapter. Instead of using these neighborhoods to perform load plan improvements, the Load Plan Neighborhood Search was used to identify load plan improvements since this technique more efficiently found improvements than restricted versions of the HLTLD formulation.

The IP-based neighborhood search heuristic follows the scheme presented in Algorithm 3. The time alloted for each solve step in the approach is discussed later in Section 4.6.

Algorithm 3: HLTLD IP-Based Neighborhood Search Scheme

```
Solve IP;
Solve Load Plan Neighorhood Search;
while the search time has not exceeded time limit T do
    if # BB selected ≤ Allowed then
        Solve Attract Freight to New BB;
        Solve Load Plan Neighorhood Search;
    end
    for every BB do
        Solve Swap BB;
        Solve Load Plan Neighorhood Search;
    end
end
Solve IP;
```

4.5 Evaluating the Impact of Trailer Cost and Dispatch Timing

In order to provide evidence that considering freight dispatch timing and trailer transportation costs enables a better set of BB terminals to be selected with respect to the total operating cost of the resulting load plan, we introduce three alternative IP formulations that can be used to select BB terminals that use a simpler model or ignore one or both of these characteristics. These additional formulations include a traditional **Hub Location** formulation, a "straw-man" **Facility Location** formulation, and a static version of the HLTLD formulation. The static version of the HLTLD formulation is modeled using the linehaul network LN instead of the time space linehaul network $TS - LN$. Each of these formulations is used to select the set of BB terminals, and in a second step the load plan is determined given the selected BB terminals. Given the set of BB terminals is identified, the corresponding load plan is used to judge the quality of the terminal choices. The quality of a load plan is measured by the cost savings ($\Delta Cost$), where cost savings are measured relative to the baseline cost when each shipment is moved direct from its origin to destination.

The set of selected BB terminals and resulting load plan using the HLTLD formulation will be compared with the BB terminals selected and corresponding load plan using each of the additional formulations. Let us now define each of the formulations considered.

4.5.1 Hub Location Formulation

The problem addressed in this study can be modeled as a multiple allocation p-hub median problem, which is to determine the set of terminals to be operated as hubs, or BB terminals, given a set of o-d freight flows. We provide a formulation for this version of the hub location problem that follows the standard form found in the literature. To formulate this problem, we will use the same notation for the linehaul network $LN = (U, L)$ and cost per trailer on a direct (i, j) represented by c_{ij}. Let K denote the set of o-d pairs considered in the problem with freight flowing between them. The weight in trailers of the freight for each commodity k between i and j is denoted q_{ij}. As before, BB_{max} is the maximum number of BB terminals that can be selected. In the hub location literature, the economies of scale gained from concentrated freight flow between hub locations are represented using a discount factor to represent the smaller cost of flow between the hub terminals when compared to the original costs. Let α denote the discount factor used for the hub-to-hub connections.

The integer program has two sets of decision variables. First, x_{ij} is a binary variable that has a value of one if terminal i is assigned to hub j and zero otherwise. Second, x_{ijkm} is a binary variable, where the value is equal to one if freight flow from i to j is routed via hubs at k and m and zero otherwise. Note that when $k = m$, both i and j are assigned to the same hub location. The cost of moving flow from i to j routed via hubs at k and m is calculated as follows: $C_{ijkm} = c_{ik} + c_{mj} + \alpha c_{km}$.

The formulation is to then minimize

$$min \sum_{i\in U} \sum_{j\in U} \sum_{k\in U} \sum_{m\in U} q_{ij} C_{ijkm} x_{ijkm}$$

subject to

$$\sum_{k\in U} \sum_{m\in U} x_{ijkm} = 1 \qquad \forall ij \in K \tag{30}$$

$$x_{ijkm} \leq x_{kk} \qquad \forall i, j, k, m \tag{31}$$

$$x_{ijkm} \leq x_{mm} \qquad \forall i, j, k, m \tag{32}$$

$$\sum_{k \in U} x_{kk} \leq BB_{max} \tag{33}$$

The objective function is to minimize the linear transportation cost based on the paths chosen to route the freight. Constraints (30) choose a single path for each origin-destination pair. Constraints (31) and (32) make sure that for each path chosen the corresponding BB terminals are also selected. Finally, constraints (33) ensures that the maximum number of BB terminals is not violated.

4.5.2 Facility Location Formulation

A facility location formulation was developed as a baseline comparison for determining a set of terminals to operate as BB terminals. This formulation represents a very simple approach to solve this strategic planning problem using a well-studied formulation. In this formulation, all inbound and outbound freight for terminal i is routed through a single hub j. Therefore, the cost of inbound and outbound transportation can be represented in the cost structure, but the hub-to-hub transportation is not captured by this model.

Using the defined linehaul network $LN = (U, L)$, let d_{ij} denote the distance between terminals i and j. Let the weight in fractional trailers of the freight for each commodity between o and d be denoted q_{od}. In this formulation, c_{ij} represents the cost of assigning terminal i to hub j. This cost is calculated as follows:

$$c_{ij} = \left(\sum_{d \in U} q_{id} + \sum_{o \in U} q_{oi} \right) * d_{ij}.$$

As before, BB_{max} is the maximum number of BB terminals that can be selected.

The integer program has two sets of decision variables. First, x_{ij} is a binary variable, where a value of one indicates that terminal i is assigned to hub j and zero otherwise. Second, z_j is a binary variable, where a value of one indicates that terminal j is operated as a BB and zero otherwise.

The formulation is to then minimize

$$min \sum_{i \in U} \sum_{j \in U} c_{ij} x_{ij}$$

subject to

$$\sum_{j \in U} x_{ij} = 1 \qquad \forall i \in U \tag{34}$$

$$x_{ij} \leq z_j \qquad \forall i, j \tag{35}$$

$$\sum_{j \in U} z_j \leq BB_{max} \tag{36}$$

The objective function is to minimize the linear cost based on the assignment of terminals to BBs. Constraints (34) choose a single BB assignment for each terminal. Constraints (35) ensure that a terminal is only assigned to a terminal that is selected to operate as a BB terminal. Finally, constraint (36) ensures that the maximum number of BB terminals is not violated.

4.5.3 Static Hub Location Load Plan Design Formulation

Because we want to be able to demonstrate the benefits of considering accurate freight dispatch timing when determining a set of BB terminals, we also developed a version of the HLTLP formulation to be solved on the static network. This means that the trailer transportation costs are calculated using the linehual network $LN = (U, L)$. The same notation is used in this formulation as in the HLTLP formulation. Now, however, variables τ_{ij} represent integer trailer flow on direct $(i, j) \in L$ instead of the integer trailer flow on arc $a \in A$.

The formulation is to then minimize

$$\sum_{(i,j) \in L} c_{ij} \tau_{ij} + \sum_{k \in K} \sum_{p \in P(k)} h_p^k x_p^k$$

subject to

$$\sum_{p \in P(k)} x_p^k = 1 \qquad \forall k \in K \tag{37}$$

$$\sum_{l \in \Delta^+(u)} y_l^d \leq 1 \qquad \forall u \in U, \forall d \in U \tag{38}$$

$$\sum_{p \in P(k): l \in p} x_p^k \leq y_l^{d(k)} \qquad \forall k \in K, \forall l \in L \tag{39}$$

$$y_l^d \leq z_{d(l)} \quad \forall d \in B, \forall l \in L : d(l) \neq d \tag{40}$$

$$\sum_{d \in B} z_d \leq BB_{max} \tag{41}$$

$$\sum_{k \in K} \sum_{p \in P(k)} w_{ij}^{kp} x_p^k \leq \tau_{ij} \qquad \forall (i,j) \in L \tag{42}$$

The objective is to minimize the transportation and handling costs. Constraints (37) ensures that one path is chosen for each o-d pair k. Constraints (38) ensure that a single outbound direct is selected for each terminal u and freight destined for terminal d. Constraints (39) ensure that a path can only be chosen for commodity k when all of its component directs are chosen. Constraints (40) ensures that a direct can only be used in an intree for a destination if the transfer terminal is selected as a BB. The constraint (41) limits the number of terminals used as BBs. And finally, constraints (42) count the number of trailers on each direct.

Now that all of the solution approaches have been defined, Table 22 provides a brief summary of each of the techniques. This table highlights the three important characteristics that need to be modeled: (1) hub-to-hub modeling, (2) trailer level costing, and (3) explicit modeling of time. For each approach, the table indicates if each of the characteristics are modeled.

Table 22: Solution Approach Summary

Approach	Hub-to-Hub Modeling	Trailer Level Costing	Explicit Modeling of Time
Facility Location			
Hub Location	X		
Static HLTLD	X	X	
HLTLD	X	X	X

4.6 Computational Results

The models and algorithms in this chapter are implemented in C++ using ILOG Concert Technology with ILOG CPLEX 12.2 as the solver. When solving IPs all parameters are set to default values unless otherwise stated; multithreading was disabled for all experiments. The experiments were run on a cluster of workstations with clockspeeds between 2.0 and 3.5 MHz all with 1.5Gb of memory.

When solving the instances in this computational study, we used a time limit of two hours for the strategic planning problem of determining the terminal roles. This time limit was chosen based on the size of the networks evaluated. Because the load plan improvement planning problem can be solved in less than ten minutes for these instances, we felt that a two hour time limit for the strategic planning problem was an adequate amount of time. If larger networks were to be considered, then the time spent solving this problem would need to be reevaluated and increased accordingly.

The IP-based neighborhood search heuristic presented in Algorithm 3 uses a time limit of 2 hours. Each IP formulation is given a 15 minute time limit to find an improving solution. Each time the Load Plan Neighborhood Search routine is run, the approach is given one minute to find load plan improvements.

For all experiments, we use cost savings ($\Delta Cost$) as the primary performance measure, where cost savings are measured in percentages relative to the baseline cost when each shipment is moved direct from its origin to destination without using any transfer terminals. The cost savings are reported for each of the improved load plans using the BB terminals selected by the approach. To perform this comparison, we run the Load Plan Neighborhood Search after the set of BB terminals is selected by a given solution approach. However, in order to run the load plan improvement step, we need a timed initial load plan for each set of BB terminals. After running the Static Hub Location Load Plan Design Formulation, we have both a set of BB terminals and a static load plan, so we have to define a timed version of the load plan. For the Hub Location and Facility Location formulations, only the set of BB terminals is defined, so we define the static load plan to be the direct load plan that transports each commodity directly from origin to destination with no intermediate transfers. Once a static load plan that dictates the flow of freight on a static network is identified, it is necessary to determine dispatch information for freight commodities on the time-space network. In this research, the initial timed load plan information was generated using an IP formulation that chooses between three timed paths for each commodity. Although an initial timed load plan can easily be developed by assigning each freight commodity to a single timed path option, we found that using this IP formulation provided better initial timed load plans. The IP formulation used to assign timed paths for each commodity is given one minute to solve, which is adequate time for the instance sizes considered in this study.

When running the Load Plan Neighborhood Search, we use a time limit of 30 minutes. For the instances evaluated in this computational study, ten minutes is adequate time to find an improved load plan solution. However, we chose to extend the allotted time to 30 minutes to ensure that the solution has reached a local optimum and is no longer finding improvements.

4.6.1 Generated Instances

For this computational study, instances were generated using data from a national LTL carrier. Because the HLTLD problem includes both terminal choices and load plan information, considering networks the size of those seen in practice by a national LTL carrier may be intractable. Therefore, since our primary goal is to provide evidence that considering freight dispatch timing and trailer transportation costs impact the choice of BB terminals, we have chosen to evaluate the performance of the proposed solution approaches on a smaller network. Our test instances are generated from a portion of the linehaul network of a national LTL carrier and the corresponding freight demands. Figure 8 contains the maps of the networks used during the computational experiments. Seven networks are considered ranging in size from 10 terminals to 50 terminals. The networks were created so that the smaller networks contain a subset of the terminals found in the larger networks.

Now that we have defined the instances for our computational results, we will begin our comparison of solution approaches with experimentation directly comparing the static and timed HLTLD formulations in Section 4.6.2. Then, in Section 4.6.3, we provide results detailing the benefits gained by tuning the parameters in CPLEX for the timed HLTLD formulation. Section 4.6.4 presents a comparison of directly solving the timed HLTLD formulation and using the IP-based local search approach. Finally, Section 4.6.5 completes the computational results section by comparing all of the solution approaches discussed.

4.6.2 Static versus Timed HLTLD Formulation

To determine the value of considering freight dispatch timing when choosing the set of terminals to operate as BB terminals, we compared the terminals selected using the static HLTLD formulation defined in Section 4.5.3 to the terminals selected using the timed HLTLD formulation defined in Section 4.3. The BB choices and corresponding load plan on a simple 10 terminal network were compared. The set of BB terminals were selected by the static or timed HLTLD formulation, in addition to the initial load plan for those terminals. Then, the Load Plan Neighborhood Search was used to find an improved load plan on the time-space network.

Each of the instances using a 10 terminal network has the same total weekly freight flows between each origin-destination pair. In the **1day-10** instance, the total weekly freight for each o-d pair is assigned at random to one day during the 5-day week. In the **2day-10** instance, the total weekly freight for each o-d pair is split equally and assigned at random to two days during the 5-day week. And finally, in the **5day-10** instance, the total weekly freight for each o-d pair is split equally across the 5-day week, meaning that the same freight flow travels every day of the week between each o-d pair.

In Table 23, the results for each HLTLD formulation are reported. This table contains results for 3 instances using a 10 terminal network considering BB_{max} set to 1, 2, or 3 terminals. For the static and timed formulation, the number constraints in the IP formulation, number of variables in the IP formulation, time in seconds to solve the IP, optimality gap of the IP, BB terminals selected, and cost savings ($\Delta Cost$) are reported. The solve time for each IP formulation is restricted to 2 hours (7200 seconds). The relative optimality gap tolerance is set to 0.50% when solving each of the IP formulations.

The results in Table 23 show that more cost savings are found when using the timed HLTLD formulation. Although the size of the timed HLTLD formulation is larger than the static HLTLD formulation with respect to both the number of variables and constraints, the set of terminals selected as BB terminals and the initial load plan defined by the timed HLTLD formulation result in greater cost savings in almost every instance. When BB_{max} is set to 1 terminal for all three instances, the static HLTLD formulation identifies a different terminal than the timed HLTLD formulation. This clearly shows that considering the freight dispatch timing allows better BB selections to be made by the HLTLD formulation. In the **1day-10** instance with BB_{max} set to 1 terminal, the static HLTLD formulation solution

Table 23: Comparison of Static vs. Timed HLTLD Formulation

		Static						Timed					
Instance	BB_{max}	# constr.	# vars	Time (secs)	Opt Gap	BBs selected	$\Delta Cost$	# constr.	# vars	Time (secs)	Opt Gap	BBs selected	$\Delta Cost$
1day-10	1	2881	1691	6.87	0.50%	MPS	-3.14%	7298	4197	0.31	0.47%	OKC	6.91%
1day-10	2	2881	1691	7200	0.82%	MPS, OKC	6.35%	7298	4197	1.38	0.20%	MPS, OKC	7.86%
1day-10	3	2881	1691	7200	0.84%	MPS, OKC, PHX	7.38%	7298	4197	1.63	0.39%	MPS, OKC, PHX	9.32%
2day-10	1	2881	1691	13.82	0.46%	MPS	9.09%	13012	6965	3.51	0.49%	OKC	22.03%
2day-10	2	2881	1691	7200	0.76%	MPS, OKC	22.36%	13012	6965	1225.88	0.50%	MPS, OKC	25.72%
2day-10	3	2881	1691	7200	1.25%	MPS, OKC, ELP	23.42%	13012	6965	1775.51	0.50%	MPS, OKC, PHX	26.32%
5day-10	1	2881	1691	4.07	0.50%	MPS	40.56%	29643	14515	7200	0.77%	OKC	46.97%
5day-10	2	2881	1691	7200	0.60%	MPS, OKC	56.71%	29643	14515	7200	4.42%	MPS, OKC	58.19%
5day-10	3	2881	1691	7200	1.17%	MPS, OKC, ELP	59.93%	29643	14515	7200	3.10%	MPS, OKC, PHX	59.34%

generates a total savings of -3.14% given the final total transportation and handling cost of the improved load plan. This negative savings (increase in transporation cost) when compared to the direct load plan is a result of the empty repositioning required from the resulting load plan. Recall that empty repositioning is ignored by the HLTLD formulation, but this is considered by the Load Plan Neighborhood Search used to find the final improved load plan for a given set of BB terminals.

The **5day-10** instance with BB_{max} set to 3 terminals is the only instance that generated a larger savings when using the static HLTLD formulation. For this instance, there is 0.59% more cost savings generated by the load plan associated with the BBs selected by the static HLTLD formulation. One should note that the **5day-10** instance is most closely related to the static case since the same freight demand occurs each day. Therefore, the static HLTLD formulation is more competitive when there are equal, daily freight demands between each o-d pair. However, the cases when BB_{max} is set to 1 and 2 terminals show that even when demand has these characteristics, the timed HLTLD formulation can still outperform the static version.

Finally, it should be noted that even when the static and timed HLTLD formulations selected the same set of BB terminals, the timed HLTLD formulation resulted in larger cost savings after the Load Plan Neighborhood Search was performed on the initial load plan. These larger cost savings can be attributed to the fact that the timed HLTLD formulation is able to generate a lower cost initial load plan given the selected set of BB terminals since it more accurately models consolidation opportunities.

4.6.3 Tuning CPLEX for the Timed HLTLD Formulation

As the number of terminals and directs in the network increases, the timed HLTLD formulation becomes more difficult to solve. Therefore, experimentation was performed to see if tuning various CPLEX parameter settings would improve the solutions found for the timed HLTLD formulation. We explored the impact of three parameter settings: MIP emphasis, dive type, and branching priorities on various variables. The MIP emphasis parameter defines if the focus of the solution search emphasizes feasibility, optimality, or a balance of the two goals. The CPLEX default setting for the MIP emphasis parameter is a balance of optimality and feasibility. The dive type parameter defines the way in which the branch and bound tree is searched, whether it be a traditional dive that never performs a probing dive, a probing dive that may be helpful in finding integer solutions, or the default setting of automatic that lets CPLEX choose when to perform a probing dive. Finally, branching priorities can be defined for each variable to instruct CPLEX to branch on integer variables with higher priority first. We tested the impact of setting branching priorities on the BB variables and intree variables.

Experimentation used three parameter settings on three instances with the same 25 terminal network. We let BB_{max} range from 4 to 7 for these three instances. Each of these instances has the same total weekly freight flows between each origin-destination pair. The three instances were created like earlier, by assigning each total o-d pair freight flow equally to one random day, two random days, or all days of the week.

Table 24: Tuning CPLEX Parameters

Instance	BB_{max}	# constraints	# variables	Default Opt Gap	Tuned Opt Gap
1day-25	4	152898	74771	12.85%	7.22%
1day-25	5	152898	74771	8.78%	7.22%
1day-25	6	152898	74771	30.90%	8.88%
1day-25	7	152898	74771	30.86%	14.81%
2day-25	4	281212	126127	55.63%	55.70%
2day-25	5	281212	126127	55.90%	32.71%
2day-25	6	281212	126127	56.44%	35.30%
2day-25	7	281212	126127	56.61%	28.37%
5day-25	4	660955	273982	80.19%	79.97%
5day-25	5	660955	273982	80.62%	80.46%
5day-25	6	660955	273982	72.82%	72.62%
5day-25	7	660955	273982	80.79%	80.72%

After experimentation, the best parameter settings were found to be a feasiblity driven MIP emphasis, a traditional dive type, and setting branching priorities on the BB variables. Table 24 provides results comparing the optimality gaps when using the default settings and the tuned parameter settings for the IP formulation. All of the IPs were given a computational time limit of two hours. This table clearly shows that the tuned parameter settings improve the optimality gap for the timed HLTLD formulation.

4.6.4 Evaluating the IP-based Local Search Approach

Even after parameter tuning, the optimzation solver still struggles to close the optimality gap for some instances. Therefore, we developed an IP-based neighborhood search heuristic to more reliably identify high-quality solutions. In this experimentation, we used the three instances from the previous experiment using the 25 terminal network and included one more instance using the 25 terminal network, **Fractional-25**. This instance has daily freight flows between each o-d pair, but the freight flows are not equal. Also, each of the freight flows is only a fractional trailer value less than a single trailer.

Table 25: Comparison of Direct IP Solve to IP-based Neighborhood Search

		Direct IP Solve			IP-based NS		
Instance	BB_{max}	Opt Gap	Initial $\Delta Cost$	Improved $\Delta Cost$	Opt Gap	Initial $\Delta Cost$	Improved $\Delta Cost$
Fractional-25	4	73.26%	6.47%	59.15%	30.90%	51.56%	61.44%
Fractional-25	5	76.81%	6.76%	60.54%	42.12%	49.17%	60.70%
Fractional-25	6	66.26%	23.04%	59.78%	36.09%	51.83%	62.01%
Fractional-25	7	66.65%	22.98%	61.18%	34.35%	51.44%	62.66%
1day-25	4	7.22%	22.49%	22.96%	5.22%	24.16%	24.19%
1day-25	5	7.22%	22.81%	22.81%	6.47%	23.69%	23.93%
1day-25	6	8.88%	21.85%	22.69%	6.16%	24.48%	24.69%
1day-25	7	14.81%	17.67%	21.36%	5.53%	24.77%	25.40%
2day-25	4	55.70%	4.70%	42.26%	19.87%	34.11%	44.01%
2day-25	5	32.71%	31.23%	44.21%	20.03%	35.73%	45.08%
2day-25	6	35.30%	27.66%	43.05%	24.65%	35.81%	43.50%
2day-25	7	28.37%	32.98%	45.37%	23.01%	37.77%	45.25%
5day-25	4	79.97%	7.25%	68.99%	33.95%	61.61%	71.32%
5day-25	5	80.46%	7.13%	69.32%	40.36%	61.32%	70.66%
5day-25	6	72.62%	32.84%	67.45%	44.45%	59.42%	70.95%
5day-25	7	80.72%	8.21%	70.22%	50.24%	64.30%	69.48%

Table 25 contains the results from the experimentation comparing directly solving the timed HLTLD formulation to the IP-based neighborhood search approach given the two hour time limit. For each instance in this table, the BB_{max}, number of constraints in the IP formulation, number of variables in the IP formulation, and information for the direct IP solve and IP-based neighborhood search. For each solution approach, the optimality gap and initial and improved cost savings ($\Delta Cost$) are reported. Since the IP-based neighborhood search does not generate a valid lower bound, the optimality gap for the IP-based neighborhood search is calculated using the lower bound found by the direct IP solve. The optimality gap compares the solution cost after solving the direct IP solve or IP-based neighborhood search to the lower bound generated by the direct IP solve. We report the initial and improved cost savings to clearly show the improvements found by improving the load plan in the second step. For all instances, the IP-based neighborhood search approach is able to close the optimality gap more than directly solving the IP formulation. And, in most of the instances, the resulting cost savings reported for the improved load plan solution is greater for the IP-based neighborhood search. There are only two runs that result in a greater cost savings when solving the IP directly. These results show that the IP-based neighorhood search more reliably generates good quality solutions.

4.6.5 Comparison of Hub Location Strategies

To show that modeling trailer transportation costs and freight dispatch timing into an approach used to select the set of terminals to operate as BB terminals aids in better decision making, we now compare the terminals selected by the timed HLTLD formulation defined in Section 4.3, the traditional hub location formulation presented in Section 4.5.1, and the faciltiy location formulation presented in Section 4.5.2. As before, the terminal set is selected by the IP formulation and then the Load Plan Neighborhood Search is used to find an improved load plan on the timed network.

Fourteen instances were used to compare the different solution approaches detailed for choosing a set of BB terminals. The first seven instances (**1day-10**, **2day-10**, **5day-10**, **Fractional-25**, **1day-25**, **2day-25**, and **5day-25**) were introduced in previous experiments. For the instances using the 10 terminal network, we still consider BB_{max} values of 1, 2, or 3 terminals. For the instances using the 25 terminal network, we still let BB_{max} range from 4 to 7 terminals. Seven new instances were used to evaluate the proposed approaches on additional freight flow patterns and larger network sizes. In the **Avg1day-25** and **Avg2day-25** instances, we use the same 25 terminal network and consider BB_{max} values ranging from 4 to 7 terminals. Both of these instances are created using the same total weekly freight flows as in the other 25 terminal instances, but now the amount of freight randomly assigned to each day is the average daily freight instead of using the total weekly freight split onto the number of days considered. This results in smaller freight flow values assigned to each day than in the previous 25 terminal network instances and a smaller total weekly freight flow for each instance. In the **5day-30**, **5day-35**, **5day-40**, **5day-45**, and **5day-50** instances, larger networks are introduced using a BB_{max} value set equal to 20% of total terminals in the network. For each of these networks, we consider freights flows between each o-d pair that are equal and repeated daily throughout a 5-day week.

Table 26: Technique Comparisons

		IP-based NS			Hub Location			Facility Location		
Instance	BB_{max}	Time (sec)	# Selected	$\Delta Cost$	Time (sec)	# Selected	$\Delta Cost$	Time (sec)	# Selected	$\Delta Cost$
1day-10	1	0.38	1	6.91%	0.08	1	5.66%	0.01	1	0.90%
1day-10	2	1.47	2	7.86%	0.09	2	3.74%	0	2	3.81%
1day-10	3	1.78	3	9.35%	0.09	2	3.74%	0.01	3	7.28%
2day-10	1	5.58	1	22.09%	0.06	1	8.25%	0.01	1	4.25%
2day-10	2	7203.09	2	29.09%	0.08	2	10.44%	0.01	2	15.86%
2day-10	3	7208.64	3	29.88%	0.08	2	10.44%	0	3	14.90%
5day-10	1	7211.61	1	48.21%	0.08	1	23.37%	0	1	20.54%
5day-10	2	7207.18	2	59.72%	0.07	2	30.94%	0.01	2	41.17%
5day-10	3	7202.83	3	60.58%	0.07	2	30.94%	0	3	47.20%
Fractional-25	4	7202.54	4	61.44%	66.85	4	59.26%	0.02	4	57.85%
Fractional-25	5	7202.13	5	60.70%	57.29	5	60.40%	0.04	5	59.85%
Fractional-25	6	7202.21	6	62.01%	66.57	5	60.40%	0.04	6	58.57%
Fractional-25	7	7202.11	7	62.66%	57.02	5	60.40%	0.03	7	58.59%
1day-25	4	7200.57	4	24.19%	84.35	4	16.43%	0.02	4	16.59%
1day-25	5	7201.07	5	23.93%	86.83	5	16.23%	0.02	5	16.17%
1day-25	6	7201.46	6	24.69%	88.6	5	16.43%	0.03	6	17.23%
1day-25	7	7200.6	7	25.40%	89.27	5	16.46%	0.02	7	19.76%
2day-25	4	7202.91	4	44.01%	66.23	4	38.99%	0.02	4	38.10%
2day-25	5	7204.02	5	45.08%	66.49	5	41.69%	0.02	5	36.85%
2day-25	6	7203.4	6	43.50%	66.27	5	41.69%	0.03	6	36.41%
2day-25	7	7204.63	7	45.25%	56.14	5	41.69%	0.02	7	40.14%
5day-25	4	7202.39	4	71.32%	57.96	4	67.63%	0.08	4	66.08%
5day-25	5	7203.74	5	70.66%	50.98	5	68.13%	0.04	5	67.81%
5day-25	6	7204.57	6	70.95%	55.15	5	68.20%	0.04	6	67.16%
5day-25	7	7203.54	7	69.48%	62.1	5	68.13%	0.04	7	66.80%
Avg1day-25	4	7200.47	4	53.4%	101.25	4	45.5%	0.04	4	38.2%
Avg1day-25	5	7200.55	5	54.0%	97.03	5	43.2%	0.04	5	40.8%
Avg1day-25	6	7200.53	6	54.7%	100.08	5	43.2%	0.03	6	39.4%
Avg1day-25	7	7200.41	7	56.7%	96.07	5	43.2%	0.03	7	39.3%
Avg2day-25	4	7200.45	4	65.1%	133.24	4	56.0%	0.03	4	53.2%
Avg2day-25	5	7200.63	5	65.5%	70.99	5	55.3%	0.02	5	53.1%
Avg2day-25	6	7200.69	6	60.5%	82.2	5	55.3%	0.02	6	52.7%
Avg2day-25	7	7200.38	7	62.5%	82.61	5	55.3%	0.02	7	57.9%
5day-30	6	7203.99	6	63.60%	171.59	6	62.62%	0.09	6	63.09%
5day-35	7	7200.57	7	68.86%	584.63	7	64.55%	0.03	7	63.91%
5day-40	8	7200.72	8	68.99%	1973.13	8	67.88%	0.03	8	66.96%
5day-45	9	7204.3	9	71.98%	2080.05	9	70.50%	0.05	9	71.16%
5day-50	10	7212.79	10	74.13%	3112.18	10	66.55%	0.18	10	67.08%

Table 26 provides the results for the all of the instances evaluated. For each solution approach in this table, the time in seconds to solve the IP formulation, number BBs selected, and cost savings ($\Delta Cost$) are reported. Table 27 reports the set of BB terminals selected for each of the instances. In all runs, the IP-based neighborhood search approach creates

Table 27: BBs Selected By Each Approach

Instance, BB_{max}	IP-based NS	Hub Location	Facility Location
1day-10, 1	OKC	PHX	MPS
1day-10, 2	MPS, OKC	CLV, PHX	MPS, PHX
1day-10, 3	MPS, OKC, PHX	CLV, PHX	CGO, HST, PHX
2day-10, 1	OKC	PHX	MPS
2day-10, 2	MPS, OKC	CLV, PHX	MPS, PHX
2day-10, 3	MPS, OKC, PHX	CLV, PHX	CGO, HST, PHX
5day-10, 1	OKC	PHX	MPS
5day-10,2	MPS, OKC	CLV, PHX	MPS, PHX
5day-10, 3	MPS, OKC, PHX	CLV, PHX	CGO, HST, PHX
Fractional-25, 4	CIN, NSH, OKC, SAN	NSH, PHX, SAN, STL	ATL, CGO, FON, GRL
Fractional-25, 5	CGO, CIN, NSH, OKC, SAN	NOL, NSH, PHX, SAN, STL	ATL, HST, PHX, SLC, STL
Fractional-25, 6	CIN, GRL, NSH, OKC, SAN, STL	NOL, NSH, PHX, SAN, STL	ATL, HST, PHX, SLC, STL, TPA
Fractional-25, 7	CIN, GRL, MSP, NSH, OKC, SAN, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, PHX, SLC, STL, TPA
1day-25, 4	GRL, NSH, SLC, STL	NSH, PHX, SAN, STL	ATL, CGO, FON , GRL
1day-25, 5	ATL, GRL, KCY, SLC, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, KCY, PHX
1day-25, 6	GRL, HST, NOL, OKC, SLC, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, KCY , PHX, TPA
1day-25, 7	CGO, FON, GRL, HST, NSH, SLC, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, FON , GRL, HST, KCY, TPA
2day-25, 4	GRL, NSH, SAN, STL	NSH , PHX, SAN, STL	ATL, CGO , FON, GRL
2day-25, 5	GRL, NSH , SAN, SLC, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, KCY, PHX
2day-25, 6	ATL, FON, GRL, KCY, NSH, SAN	NOL, NSH, PHX, SAN , STL	ATL, CGO , HST , KCY, PHX, TPA
2day-25, 7	ATL, GRL, KCY, NSH, OKC, SAN, STL	NOL, NSH, PHX, SAN , STL	ATL, CGO, FON , GRL, HST, KCY, TPA
5day-25, 4	FON, GRL, KCY , STL	NSH, PHX, SAN, STL	ATL, CGO, FON, GRL
5day-25, 5	GRL, KCY, NSH, SAN , STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, KCY, PHX
5day-25, 6	GRL, MSP, NSH, OKC, SAN , STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, KCY, PHX, TPA
5day-25, 7	CGO, GRL, KCY, NSH, OKC, SAN, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, FON, GRL, HST, KCY, TPA
Avg1day-25, 4	ATL, ELP, SLC, STL	NSH, PHX, SAN, STL	ATL, CGO, FON, GRL
Avg1day-25, 5	ATL, DEN, GRL, NSH, SLC	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, KCY, PHX
Avg1day-25, 6	ATL, JAX, NSH, OKC, SLC, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, KCY, PHX, TPA
Avg1day-25, 7	ATL, DEN, GRL, OKC, PHX, SLC, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, FON, GRL, HST, KCY, TPA
Avg2day-25, 4	ATL, FON, OKC, STL	NSH, PHX, SAN, STL	ATL, CGO, FON, GRL
Avg2day-25, 5	ATL, NOL, OKC, SLC, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, KCY, PHX
Avg2day-25, 6	ATL, GRL, NOL, OKC, SLC, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, HST, KCY, PHX, TPA
Avg2day-25, 7	ATL, GRL, MSP, NOL, OKC, PHX, STL	NOL, NSH, PHX, SAN, STL	ATL, CGO, FON, GRL, HST, KCY, TPA
5day-30, 6	KCY, MSP, PHX, SAN, SPT, STL	BHM, IND, LVS, MSP, PHX, SPT	ATL, CGO, GRL, HST, LVS, SEA
5day-35, 7	MSP, NSH, ODS, OKC, SEA, SPT, STL	BHM, IND, LVS, MSP, PHX, SAN, SPT	ATL, CGO, FON, GRL, HST, SEA, TPA
5day-40, 8	DEN, DSM, FON, KCY, LVS, PHX, SPF, SPT	BHM, DSM, IND, MSP, SPF, SPT, STL, TFT	ATL, CGO, FON, GRL, HST, KCY, SEA, TPA
5day-45, 9	DSM, GRL, KCY, PHX, SAN, SPF, SPT, STL, WTK	BHM, DSM, IND, KNX, MSP, SPF, SPT, STL, TFT	ATL, CGO, FON, GRL, HST, KCY, NSH, SEA, TPA
5day-50, 10	DSM, GRL, KCY, MSP, PHX, SAN, SPF, SPT, STL, WTK	BHM, DSM, IND, KNX, MSP, PHX, SPF, SPT, STL, TFT	ATL, CGO, DEN, FON, GRL, HST, KCY, NSH, SEA, TPA

the load plan with the largest cost savings when compared to the load plans created using the terminals selected by the Hub Location and Facility Location formulations.

It should be noted that both the Hub Location and Facility Location formulations require significantly less time to select the set of BB terminals than the timed HLTLD formulation, which can be attributed to the fact that both formulations are significantly smaller since they do not incorporate freight dispatch timing and trailer transportation costs. Because both formulations ignore freight dispatch timing, the same BB terminals are selected for every instance evaluated with the same total weekly freight flow even though the daily freight flows differ in each instance. For example, the **1day-25**, **2day-25**, and **5day-25** instances have the same total weekly freight between each o-d pair but the freight demands are requested on one, two, and five days, respectively. These instances were designed this way to highlight the fact that both the Hub Location and Facility Location formulations only consider weekly aggregate freight flows when choosing the set of BB terminals.

The results in Tables 26 and 27 also show that for some instances of the Hub Location formulation only a subset of the allowed number of BB terminals are selected. For example, in the 25 terminal networks the Hub Location formulation only selects 5 BB terminals when BB_{max} is set to 6 or 7 terminals. For the instances evaluated, this trend happens when BB_{max} exceeds 20% of the total terminal count for the network considered. Clearly, the results from the other two approaches show that there are benefits to considering additional BB terminals to transfer freight.

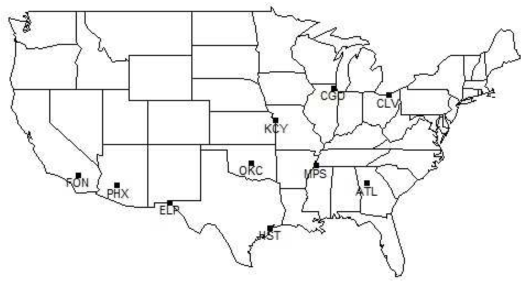
The instances with o-d freight flows only occurring on one day (*e.g.*, the **1day-10**, **1day-25**, and **Avg1day-25** instances) provide the best evidence that the consideration of freight dispatch timing enables better decision making when selecting the set of BB terminals, thus resulting in larger cost savings for the corresponding load plan. For these instances, we observe the largest differences in cost savings when comparing all three approaches. The IP-based neighborhood search approach generates notably larger cost savings, while the Hub Location and Facility Location approaches generate more comparable cost savings. Table 27 shows that the BB terminals selected for every instance are different, which strongly impacts the opportunities for potential consolidation. Table 28 reports the path breakdowns for each of the solutions for the one day freight instances. In this table, the percentage of o-d pairs sent directly from origin to destination and the percentage of those sent through one or more transfer terminals are reported for each of the solution approaches. The percentages reported in this table show that for most instances the IP-based neighborhood search selects BB terminals that allow a larger percentage of the o-d pairs to be sent through transfer terminals than the other two approaches, thus resulting in lower cost load plans. These instances clearly depict that considering weekly aggregate freight flows overestimates the potential for freight consolidation.

Table 28: Technique Comparisons: Number of Transfers

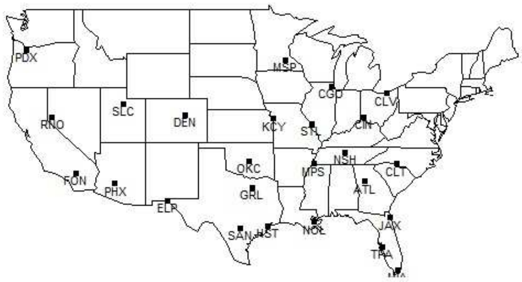
		IP-based NS		Hub Location		Facility Location	
Instance	BB_{max}	% Direct	% $\geq$ 1 Transfers	% Direct	% $\geq$ 1 Transfers	% Direct	% $\geq$ 1 Transfers
1day-10	1	70.0	30.0	88.9	11.1	83.3	16.7
1day-10	2	63.3	36.7	87.8	12.2	80.0	20.0
1day-10	3	64.4	35.6	87.8	12.2	82.2	17.8
1day-25	4	40.7	59.3	47.1	52.9	48.7	51.3
1day-25	5	42.2	57.6	45.1	54.9	48.9	51.1
1day-25	6	43.2	56.8	45.6	54.4	50.9	49.1
1day-25	7	42.1	57.9	45.6	54.4	45.1	54.9
Avg1day-25	4	27.4	72.6	27.5	72.5	29.5	70.6
Avg1day-25	5	28.0	72.0	27.0	73.0	29.7	70.3
Avg1day-25	6	26.7	73.3	27.0	73.0	31.1	68.9
Avg1day-25	7	24.0	76.0	27.0	73.0	30.9	69.1

Although the focus of this study is to provide evidence that considering freight dispatch timing and trailer transportation costs in a solution approach aids in better decision making, it is interesting that the Facility Location solution approach performs competitively with the Hub Location solution approach for many of the instances considered. The Facility Location formulation ignores the fact that total freight flow is comprised of a set of individual movements from a specific origin to a specific destination. These freight flows are indirectly incorporated into the cost structure used in the formulation, but the cost structure does not capture the freight flow through the entire network since none of the BB-to-BB freight flow costs are considered. The Hub Location formulation considers the BB-to-BB freight flow and incentivizes freight flowing on these connections with a discount factor. Given that the Hub Location formulation is tailored to identifying BB terminals using o-d freight flows, it is interesting that it does not consistently outperform the Facility Location formulation proposed. This inconsistent performance can potentially be attributed to the fact that neither of the formulations consider trailer flows and do not capture the cost structure used in the Load Plan Neighborhood Search that evaluates the quality of the BB terminal choices. Therefore, these experiments have also provided evidence that traditional hub location formulations may not represent realistic models for many transportation problems and demonstrated one drawback when using those formulations.

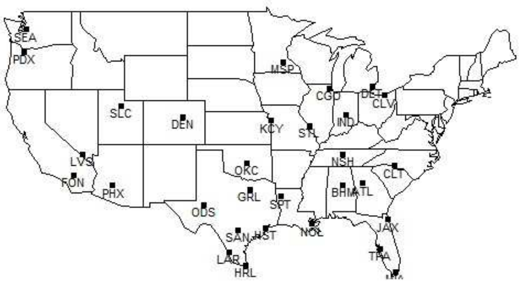
Upon further inspection of the IP-based neighborhood search approach for the larger networks, it is clear that this approach has room for improvement and performs better on networks with 25 terminals or less. For the networks with 30 terminals or more, there were no changes throughout the solution time in the set of BB terminals selected. For the smaller networks, the approach has the ability to change the set of BB terminals selected to find an improving set. This behavior is not seen in the instances with larger networks. However, it should be noted that even the initial set of BB terminals selected using this approach outperform the BB terminals selected by the Hub Location and Facility Location formulations, thus providing evidence for better decision making when freight dispatch timing and trailer transportation costs are incorporated into a solution method for choosing the set of BB terminals.



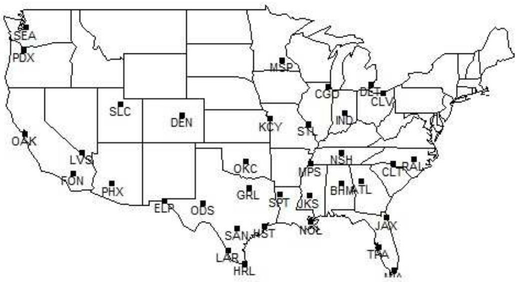
(a) 10 Terminal Network



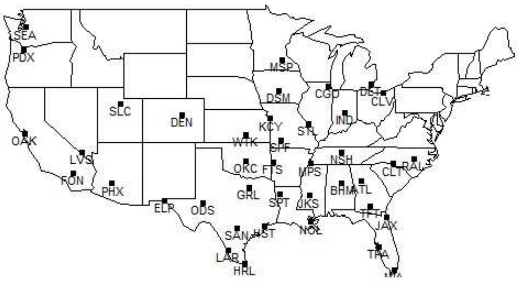
(b) 25 Terminal Network



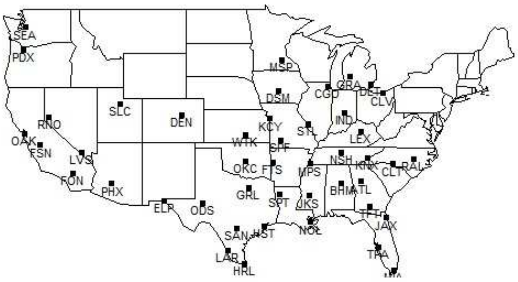
(c) 30 Terminal Network



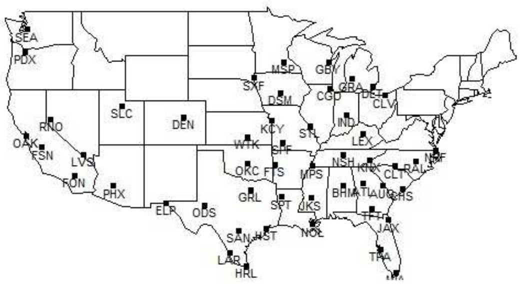
(d) 35 Terminal Network



(e) 40 Terminal Network



(f) 45 Terminal Network



(g) 50 Terminal Network

Figure 8: Terminal Networks

CHAPTER V

CONCLUSIONS AND FUTURE RESEARCH

This dissertation proposed solution approaches using both heuristic and exact optimization techniques to address strategic, tactical, and operational planning problems arising in freight consolidation networks.

Chapter 2 investigated and developed integer programming-based solution approaches for a shipper pickup and delivery planning problem faced by many large retailers. The challenge in this planning problem was to exploit as much as possible negotiated truckload lane rates and to judiciously make use of routes through crossdock facilities to consolidate shipments. The solution approach generated a set of routes that are likely to be part of high-quality solutions and then used an integer program to choose a low-cost set of routes fulfilling all orders. An IP-based local search scheme in which a sequence of restricted integer programs are solved was shown to be effective in producing high-quality solutions. The size of the integer programs grows steadily as the number of orders, and thus the set of candidate routes, increases. For the real-life instances that motivated this research, generating a complete set of candidate routes *a priori* and solving the resulting integer program using the presented search scheme worked well. However, if instances with substantially more orders needed to be solved, the approach may no longer be effective. For this reason, a natural extension for this work is to investigate column generation techniques or metaheuristic search approaches that do not rely on integer programming. These alternative techniques may provide a more robust approach for larger problem instances that were not evaluated in this study.

Although the primary value of the technology presented in Chapter 2 was to reduce transportation costs, the technology can potentially be used in support of lane rate negotiations as well. The lane rates, and the flexibility to add additional stops, have a tremendous impact on the overall costs of the shipment plan. By negotiating advantageous lane rates in strategic areas, significant cost savings can likely be achieved. Modifying the developed technology to identify lanes and lane rate structures that are most valuable in terms of reducing total costs is one of the key research challenges we plan to address in the future.

Chapter 3 considered the load plan design problem for a large-scale LTL transportation network. We developed an IP-based local search approach that generated neighborhoods to evaluate by targeting a single direct for freight attraction or reduction that impacted multiple destination trees. Using data from a large US LTL carrier, we performed a computational study that provided results showing that the approach can find load plans with costs 6 to 7% lower than those used in practice. The developed approach also finds comparable cost savings to previously proposed methods in notably less time. One venue for future work is to extend the approach presented in this chapter to encompass the idea of day-differentiated load plans, as introduced by Erera et al. [19]. By allowing the load plan to change day-by-day, there may be opportunity for more tailored freight routing based on freight flow patterns specific to days of the week. The load plan design technology could also benefit by the incorporation of more accurate cost estimation techniques, for example, those proposed by Erera et al. [20].

At the end of Chapter 3, we introduced initial ideas for creating lower bounds for the load plan design problem. The bounds discussed fouced on using the arc-based load plan design formulation as a starting point for generating lower bounds. Initial experimentation was performed on a small 10 terminal network to test some of the ideas proposed, but a more thorough investigation needs to be completed to determine if any of the lower bounds can be computed efficiently for larger networks used by large US LTL carriers. Future research addressing the load plan design problem should investigate lower bounds because the current literature does not contain much work considering this topic. By developing stronger lower bounds for the problem, the quality of the solution approaches could be more deeply understood and not just compared to load plans that are operated in practice. In addition, strong lower bounds could potentially lead to the development of improved load plan design solution approaches.

Chapter 4 extended the load plan design problem to consider terminal roles, thus introducing a new variant of a hub location problem. The proposed solution approach incorporated the approach developed for the load plan design problem into a new IP-based local search framework that also considered restrictions of the Hub Location Load Plan Design formulation to target improving changes in the set of BB terminals selected. The IP-based neighborhood search solution approach was compared with directly solving the IP formulation and shown to more reliably produce high-quality solutions. We also introduced Hub Location and Facility Location formulations to depict the benefits of incorporating trailer transportation costs and freight dispatch timing into the solution approach used to select the set of BB terminals. Computational experiments showed that the IP-based neighborhood search approach selected better BB terminals resulting in a lower cost load plan for all of the instances tested.

This work has created a successful approach for selecting a set of BB terminals and identifying the corresponding load plan on small networks, but it is not clear that this approach will scale to large networks like the one considered in Chapter 3. Therefore, one important venue for future research is to elaborate on the current approach to make it effective for larger networks. In this study, we targeted changes to the set of BB terminals using a restricted version of the Hub Location Load Plan Design formulation. Because this formulation grows very fast as the size of the network and number of commodities increases, it would be beneficial to investigate alternative ways of determining which terminals to consider as part of the set of selected BB terminals. A successful methodology may use an approach incorporating tabu search techniques as a way to change the search mechanism used to choose BB terminals. A tabu search approach could allow a more diverse search of appropriate BB terminals to be evaluated during the solution time. Another option may be to consider alternative neighborhoods used to evaluate the selection of BB terminals that can be evaluated heuristically or with a tailored IP formulation that is easier to solve than the restricted IP formulations used in this work.

REFERENCES

[1] AHUJA, R., ERGUN, Ö., ORLIN, J., and PUNNEN, A., "A survey of very large-scale neighborhood search techniques," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 75–102, 2002.

[2] ALUMUR, S. and KARA, B., "Network hub location problems: The state of the art," *European Journal of Operational Research*, vol. 190, pp. 1–21, Oct. 2008.

[3] ARCHETTI, C., SPERANZA, M. G., and SAVELSBERGH, M. W. P., "An optimization-based heuristic for the split delivery vehicle routing problem," *Transportation Science*, vol. 42, pp. 22–31, 2008.

[4] ARMACOST, A., BARNHART, C., WARE, K., and WILSON, A., "UPS optimizes its air network," *Interfaces*, vol. 34, no. 1, pp. 15–25, 2004.

[5] A.T.A, "American trucking associations trucking and the economy. 2007-2008 edition," 2008. http://www.truckline.com/Newsroom/Trucks Are/Trucking and the Economy.pdf.

[6] BARCOS, L., RODRÍGUEZ, V., ÁLVAREZ, M., and ROBUSTÉ, F., "Routing design for less-than-truckload motor carriers using ant colony optimization," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 3, pp. 367–383, 2010.

[7] BARNHART, C., JIN, H., and VANCE, P., "Railroad blocking: A network design application," *Operations Research*, vol. 48, no. 4, pp. 603–614, 2000.

[8] BARNHART, C., KRISHNAN, N., KIM, D., and WARE, K., "Network design for express shipment delivery," *Computational Optimization and Applications*, vol. 21, no. 3, pp. 239–262, 2002.

[9] BARNHART, C. and SCHNEUR, R., "Air network design for express shipment service," *Operations Research*, vol. 44, no. 6, pp. 852–863, 1996.

[10] CAMPBELL, J. F., "Hub location for time definite transportation," *Computers & Operations Research*, vol. 36, pp. 3107–3116, Dec. 2009.

[11] CHEN, P., GUO, Y., LIM, A., and RODRIGUES, B., "Multiple crossdocks with inventory and time windows," *Computers & Operations Research*, vol. 33, pp. 43–63, Jan. 2006.

[12] CHEUNG, R. and MURALIDHARAN, B., "Dynamic routing for priority shipments in ltl service networks," *Transportation science*, vol. 34, no. 1, pp. 86–98, 2000.

[13] CORTÉS, C., MATAMALA, M., and CONTARDO, C., "The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method," *European Journal of Operational Research*, vol. 200, pp. 711–724, Feb. 2010.

[14] Crainic, T., "Service netowrk design in freight transportation," *European Journal of Operational Research*, vol. 122, no. 2, pp. 272–288, 2000.

[15] Crainic, T. and Rousseau, J., "Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem," *Transportation Research Part B: Methodological*, vol. 20, no. 3, pp. 225–242, 1986.

[16] Cunha, C. B. and Silva, M. R., "A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil," *European Journal of Operational Research*, vol. 179, pp. 747–758, June 2007.

[17] da Graça Costa, M., Captivo, M. E., and Clímaco, J. a., "Capacitated single allocation hub location problemA bi-criteria approach," *Computers & Operations Research*, vol. 35, pp. 3671–3695, Nov. 2008.

[18] De Franceschi, R., Fischetti, M., and Toth, P., "A new ilp-based refinement heuristic for vehicle routing problems," *Mathematical Programming B*, vol. 105, pp. 471–499, 2006.

[19] Erera, A., Hewitt, M., Savelsbergh, M., and Zhang, Y., "Improved load plan design through integer programming based local search," *Transportation Science*, 2012.

[20] Erera, A., Hewitt, M., Savelsbergh, M., and Zhang, Y., "Creating schedules and computing operating costs for ltl load plans," *Computers & Operations Research*, 2012.

[21] Franceschi, R. D., Fischetti, M., and Toth, P., "A new ILP-based refinement heuristic for vehicle routing problems," *Mathematical Programming B*, vol. 105, pp. 471–499, 2006.

[22] Franceschi, R., Fischetti, M., and Toth, P., "A new ilp-based refinement heuristic for vehicle routing problems," *Mathematical Programming*, vol. 105, no. 2, pp. 471–499, 2006.

[23] Grünert, T. and Sebastian, H., "Planning models for long-haul operations of postal and express shipment companies," *European Journal of Operational Research*, vol. 122, no. 2, pp. 289–309, 2000.

[24] Hewitt, M., Nemhauser, G., and Savelsbergh, M., "Combining exact and heuristic approaches for the capacitated fixed charge network flow problem," *INFORMS Journal on Computing*, vol. 22, pp. 314–325, 2009.

[25] Hewitt, M., Nemhauser, G., and Savelsbergh, M., "Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem," *INFORMS Journal on Computing*, vol. 22, no. 2, pp. 314–325, 2010.

[26] Hwang, J. and Kim, S., "An integer programming-based local search for large-scale maximal covering problems," *International Journal*, vol. 3, pp. 837–843, 2011.

[27] Hwang, J., Park, S., and Kong, I., "An integer programming-based local search for large-scale multidimensional knapsack problems," *International Journal*, vol. 3, no. 6, pp. 2257–2264, 2011.

[28] Jansen, B., Swinkels, P., Teeuwen, G., van Antwerpen de Fluiter, B., and Fleuren, H., "Operational planning of a large-scale multi-modal transportation system," *European Journal of Operational Research*, vol. 156, no. 1, pp. 41–53, 2004.

[29] Jarrah, A., Johnson, E., and Neubert, L., "Large-scale, less-than-truckload service network design," *Operations Research*, vol. 57, no. 3, pp. 609–625, 2009.

[30] Jayaraman, V. and Ross, A., "A simulated annealing methodology to distribution network design and management," *European Journal of Operational Research*, vol. 144, pp. 629–645, Feb. 2003.

[31] Kara, B. and Tansel, B., "The latest arrival hub location problem," *Management Science*, vol. 47, no. 10, pp. 1408–1420, 2001.

[32] Kim, D., Barnhart, C., Ware, K., and Reinhardt, G., "Multimodal express package delivery: A service network design application," *Transportation Science*, vol. 33, no. 4, pp. 391–407, 1999.

[33] Lee, Y., Jung, J., and Lee, K., "Vehicle routing scheduling for cross-docking in the supply chain," *Computers & Industrial Engineering*, vol. 51, pp. 247–256, Oct. 2006.

[34] Li, X., Lim, A., and Miao, Z., "Reducing transportation costs in distribution networks," *Advances in Applied Artificial Intelligence*, vol. 4031/2006, pp. 1138–1148, 2006.

[35] Liao, C., Lin, Y., and Shih, S., "Vehicle routing with cross-docking in the supply chain," *Expert Systems with Applications*, vol. 37, pp. 6868–6873, Oct. 2010.

[36] Mitrovic-Minic, S. and Laporte, G., "The pickup and delivery problem with time windows and transshipment," *INFOR*, vol. 44, no. 3, pp. 217–227, 2006.

[37] Mues, C. and Pickl, S., "Transshipment and time windows in vehicle routing," in *8th International Symposium on Parallel Architetures, Algorithms and Networks*, pp. 113–119, 2005.

[38] Newton, H., Barnhart, C., and Vance, P., "Constructing railroad blocking plans to minimize handling costs," *Transportation Science*, vol. 32, no. 4, pp. 330–345, 1998.

[39] Nozick, L. and Morlok, E., "A model for medium-term operations planning in an intermodal rail-truck service," *Transportation research part a: policy and practice*, vol. 31, no. 2, pp. 91–107, 1997.

[40] Powell, W. B., "A local improvement heuristic for the design of less-than-truckload motor carrier networks," *Transportation Science*, vol. 20, no. 4, pp. 246–257, 1986.

[41] Powell, W. B. and Koskosidis, I. A., "Shipment routing algorithms with tree constraints," *Transportation Science*, vol. 26, no. 3, pp. 230–245, 1992.

[42] Powell, W. B. and Sheffi, Y., "The load planning problem for motor carriers: Problem description and proposed solution approach," *Transportation Science*, vol. 17A, no. 6, pp. 471–480, 1983.

[43] Powell, W. B. and Sheffi, Y., "Design and implementation of an interactive optimization system for newtork design in the motor carrier industry," *Operations Research*, vol. 37, pp. 12–29, 1989.

[44] Salari, M. and Naji-Azimi, Z., "An integer programming-based local search for the covering salesman problem," *Computers & Operations Research*, vol. 39, no. 11, pp. 2594–2602, 2012.

[45] Savelsbergh, M. and Song, J., "An optimization algorithm for inventory routing with continuous moves," *Computers and Operations Research*, vol. 35, pp. 2266–2282, 2008.

[46] Savelsbergh, M. and Sol, M., "The general pickup and delivery problem," *Transportation Science*, vol. 29, no. 1, pp. 17–29, 1995.

[47] Sung, C. and Song, S., "Integrated service network design for a cross-docking supply chain network," *Journal of the Operational Research Society*, vol. 54, pp. 1283–1295, Dec. 2003.

[48] TruckInfo.net, "Trucking statistics." http://www.truckinfo.net/trucking/stats.htm, February 2012.

[49] Üster, H. and Agrahari, H., "An integrated load-planning problem with intermediate consolidated truckload assignments," *IIE Transactions*, vol. 42, no. 7, pp. 490–513, 2010.

[50] Üster, H. and Maheshwari, N., "Strategic network design for multi-zone truckload shipments," *IIE Transactions*, vol. 39, no. 2, pp. 177–189, 2007.

[51] Wen, M., Larsen, J., Clausen, J., and Laporte, G., "Vehicle routing with cross-docking," *Journal of the Operational Research Society*, vol. 60, pp. 1708–1718, 2009.

[52] Wieberneit, N., "Service network design for freight transportation: a review.," *OR Spectrum*, vol. 30, no. 1, pp. 77 – 112, 2008.

[53] Yaman, H., Kara, B., and Tansel, B., "The latest arrival hub location problem for cargo delivery systems with stopovers," *Transportation Research Part B:*, vol. 41, no. 8, p. 906, 2007.

[54] Zhang, Y., *Advances in LTL Load Plan Design.* PhD thesis, 2010.

[55] Zhu, E., *Scheduled service network design for integrated planning of rail freight transportation.* PhD thesis, 2011.